University of Alberta

Focused Co-citation: Improving the Retrieval of Related Pages on the Web

by

Gabriela Moise

A thesis submitted to the Faculty of Graduate Studies and Research in partial fulfillment of the requirements for the degree of **Master of Science**.

Department of Computing Science

Edmonton, Alberta
Fall 2003

National Library
of Canada

Acquisitions and
Bibliographic Services

395 Wellington Street
Ottawa ON K1A 0N4
Canada

Bibliothèque nationale
du Canada

Acquisisitons et
services bibliographiques

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file Votre référence
ISBN: 0-612-87839-2
Our file Notre référence
ISBN: 0-612-87839-2

The author has granted a non-exclusive licence allowing the National Library of Canada to reproduce, loan, distribute or sell copies of this thesis in microform, paper or electronic formats.

The author retains ownership of the copyright in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

L'auteur a accordé une licence non exclusive permettant à la Bibliothèque nationale du Canada de reproduire, prêter, distribuer ou vendre des copies de cette thèse sous la forme de microfiche/film, de reproduction sur papier ou sur format électronique.

L'auteur conserve la propriété du droit d'auteur qui protège cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou aturement reproduits sans son autorisation.

Canada

University of Alberta

Library Release Form

Name of Author: Gabriela Moise

Title of Thesis: Focused Co-citation: Improving the Retrieval of Related Pages on the Web

Degree: Master of Science

Year this Degree Granted: 2003

Permission is hereby granted to the University of Alberta Library to reproduce single copies of this thesis and to lend or sell such copies for private, scholarly or scientific research purposes only.

The author reserves all other publication and other rights in association with the copyright in the thesis, and except as herein before provided, neither the thesis nor any substantial portion thereof may be printed or otherwise reproduced in any material form whatever without the author's prior written permission.

_____

Date: June 19th 2003

University of Alberta

Faculty of Graduate Studies and Research

The undersigned certify that they have read, and recommend to the Faculty of Graduate Studies and Research for acceptance, a thesis entitled **Focused Co-citation: Improving the Retrieval of Related Pages on the Web** submitted by Gabriela Moise in partial fulfillment of the requirements for the degree of **Master of Science**.

Jörg Sander
Co-Supervisor

Davood Rafiei
Co-Supervisor

Eleni [illegible]

Witold Pedrycz

Date: May 16th 2003

Abstract

This thesis studies the problem of effectively finding related pages on the Web, where given the URL of a page, one wants to find other pages that are on the same topic. This is a both simple and natural way of searching for resources without being forced to formulate a search query using some keywords.

A number of problems that often arise on the Web and affect the precision of algorithms that use the link structure of the Web to find related pages are identified. To address these problems, several new notions of "focus" of a collection of links are proposed and embedded within the Co-citation algorithm. The goal is that, when searching for related pages, an algorithm should give more focused collections of links a higher influence on the final ranking than less focused collections. Our experiments show that the "focused" versions of Co-citation outperform the unfocused version.

To my husband, who constantly helped and supported me.

Acknowledgements

I would like to thank to my supervisors, Dr. Jörg Sander and Dr. Davood Rafiei for their constant help and support.

I would also like to thank to the members of my committee, Dr. Eleni Stroulia and Dr. Witold Pedrycz for their insightful feedback.

Contents

1 Introduction 1
- 1.1 Motivation . . . 1
- 1.2 Related Work . . . 3
- 1.3 Our contribution . . . 6

2 Related Work 8
- 2.1 WWW and Web Mining . . . 8
- 2.2 Web Structure Mining . . . 10
  - 2.2.1 Studies of the Web structure . . . 12
  - 2.2.2 Relevance of Web pages . . . 13
  - 2.2.3 Web communities . . . 19
- 2.3 "Related pages" algorithms . . . 20
  - 2.3.1 Companion algorithm . . . 21
  - 2.3.2 Co-citation algorithm . . . 24
- 2.4 Work related to "pagelets" . . . 27
- 2.5 Work related to "nepotistic" links . . . 29
- 2.6 Work related to "near-duplicate" Web pages . . . 30
- 2.7 Work related to evaluation strategies . . . 30

3 Problems with Co-citation 32
- 3.1 Original Co-citation . . . 33
- 3.2 Navigational Links . . . 34
- 3.3 Near-duplicate pages . . . 40
- 3.4 "Unfocused" collection of links . . . 44
- 3.5 The need for "focus" . . . 47
- 3.6 Preprocessing - Discussion . . . 48

4 Focused Co-citation 51
- 4.1 Ranking Function . . . 52
- 4.2 LinkFocus . . . 54
- 4.3 Content Focus . . . 58
- 4.4 HybridFocus . . . 67
- 4.5 Summary . . . 70

5 Experimental Evaluation **71**
5.1 Implementation 72
5.2 Evaluation Strategy 73
5.3 Experimental Results 76
5.4 Statistical Significance 82
5.5 Discussion 87
5.6 Manual Evaluation 89

6 Conclusions and Future Work **91**
6.1 Lessons Learned 92
6.2 Directions for Future Work 92

Bibliography **93**

List of Figures

2.1 Companion algorithm 22
2.2 Companion: building the vicinity graph 23
2.3 Co-citation algorithm 25
2.4 Co-citation: building the vicinity graph 26
2.5 Pagelet extraction algorithm 29

3.1 Standard navigational panel 34
3.2 Navigational links from a "more specific" to a "less specific" domain 36
3.3 Homepages - navigational links 38
3.4 Navigational links elimination 39
3.5 Navigational links elimination and near-duplicates contraction. 44
3.6 Preprocessed Co-citation. 46

4.1 LinkBased Focus 54
4.2 ContentBased Focus 59
4.3 HybridFocus: joined Link and Content Focus 69
4.4 Focused Co-citation 70

5.1 Collapsed Open Directory 75
5.2 Preprocessed data: Precision at 10 76
5.3 Preprocessed data: Average Precision 77
5.4 Preprocessed data: Precision at R 77
5.5 Un-Preprocessed data: Precision at 10 78
5.6 Un-Preprocessed data: Average Precision 79
5.7 Un-Preprocessed data: Precision at R 79
5.8 Preprocessed Data - Reduced Set: Precision at 10 81
5.9 Preprocessed Data - Reduced Set: Average Precision 82
5.10 Un-Preprocessed Data - Reduced Set: Precision at 10 82
5.11 Un-Preprocessed Data - Reduced Set: Average Precision . . . 83
5.12 Selecting keywords: initial set 87
5.13 Automatic selection of keywords: Precision at 10 89
5.14 Automatic selection of keywords: Average Precision 90
5.15 Manual selection of keywords: Precision at 10 90
5.16 Manual selection of keywords: Average Precision 90

List of Tables

3.1 Original Co-citation - results 33
3.2 Original Co-citation - Results after navigational links elimination 40
3.3 Original Co-citation - Results after navigational links elimination and near-duplicates contraction 45
3.4 Original Co-citation - Results after navigational links elimination and duplicates contraction and pagelet extraction 47
3.5 Example: Original Co-citation 49
3.6 Example: Preprocessed Co-citation 49

4.1 LinkFocus 57
4.2 Frequency of keywords 63
4.3 ContentFocus 67

5.1 Evaluated Algorithms 72
5.2 Preprocessed data: Overlap between methods 80
5.3 Un-preprocessed data: Overlap between methods 80
5.4 Preprocessed Data: Sign test and Wilcoxon sum of ranks test 83
5.5 Preprocessed Data: $\alpha = 0.05$ 84
5.6 Preprocessed Data: $\alpha = 0.01$ 85
5.7 Un-Preprocessed Data: Sign test and Wilcoxon sum of ranks test 85
5.8 Un-Preprocessed Data: $\alpha = 0.05$ 86
5.9 Un-Preprocessed Data: $\alpha = 0.01$ 86
5.10 Selecting keywords: "computer" and "science" 88
5.11 Selecting keywords: "ubc" and "department" 88

Chapter 1

Introduction

This thesis addresses the problem of effectively finding related pages on the World Wide Web (WWW, Web), where given the URL of a page, we want to find other pages that are related or similar to the given page in some context. Given a Web page, by related Web pages, we understand those pages that address the same topic (i.e., subject of interest) as the original page. For example, given the *homepage* of a computer science department, *www.cs.ubc.ca*, we deem as related *homepages* of other computing science departments.

1.1 Motivation

Traditional search engines take as input a query, composed of a set of keywords, and output (ideally) a set of relevant pages. Using such search engines to locate resources on the Web is useful in many cases. However, when expressing their information need by a set of keywords, users often commit errors, which in turn will determine the search engine to return a set of pages that are only marginally relevant to the users' interests. For example, let us consider the case when a user is interested in locating information about Fuji, the famous Japanese volcano. When typing the query "Fuji" on Google [20], the majority of the high-ranked results are about photography and television, Fuji being also a renowned company in these areas (the Mount Fuji's *homepage* is ranked 30^{th}). There are also cases when the results returned by the search engines are not relevant (or only tangentially relevant) to the users' interests because of the searching or ranking strategy of the search engines themselves. For instance,

when searching for "hub" pages, i.e. collection of links relevant to a specific topic, Google has difficulties in returning relevant results in the first positions. This is because the way Google ranks its results, i.e. using PageRank (see chapter 2). For example, let us consider the case when a user wants to find a list of scientists of the Renaissance period. When using diverse keywords combination, such as "renaissance scientists " or "renaissance scientists list", Google does not return any *collection* of such scientists, but rather pages relevant to individual personalities of the time (or irrelevant pages).

Finding related Web pages is a both simple and natural way of searching for resources without the need to formulate a search query using some keywords. The input to the search process is not a set of keywords, but the URL of a page of interest and the output is a set of related Web pages. Obviously, it is assumed that the user has already found a page of interest. This could be done by browsing some of the existing directories on the Web, such as Open Directory [30], Yahoo! [42] that try to classify Web pages under a set of categories, using a tree-like structure.

Searching for related pages has several interesting applications. One application is the *identification of topicality*, i.e. identification of a set of pages that are relevant to some degree, ranging from pages highly relevant to pages marginally relevant, to a given topic. Starting with one page on a topic of interest, the tool that retrieves related pages would allow the user to identify pages that are highly relevant to the given topic. Applying the searching tool repeatedly could further extend this collection of related pages: each of the pages obtained in the current step could produce a set of related pages. After several steps, the pages obtained will be only tangentially relevant to the starting topic.

Another application is the *clustering of Web pages*. Assuming that a degree of relatedness is defined, relatedness can be seen as a similarity measure, which can be used in clustering algorithms to identify "natural" groups of pages. Those groups may often correspond to or indicate interesting properties of real world entities, such as user groups who share similar interests.

1.2 Related Work

Since the advent of the Web, a novel area came into shape, Web mining, which could be generally described as the process of extracting patterns from Web data through content mining, structure mining and usage mining. Recent research in the Web mining field came to acknowledge that the hyperlink structure of the Web could be a very valuable tool for locating information ([15], [34], [24], [31]). The basic assumption is that if there is a link from page p to page q, then the author of page p recommends page q, and often links connect related pages.

Web structure mining or link analysis has been used for different purposes: 1) to study the Web structure, 2) to compute the relevance of Web pages or 3) to identify communities on the Web.

Given the fact that link analysis is essentially about graph structure, several models for the Web have been proposed ([33], [8], [25]). One of these models for the Web is most often encountered. This model shapes the Web as a directed graph, where the nodes of the graph are the Web pages and the edges are the existing hyperlinks between Web pages: if page p is pointing to page q (i.e. there is a hyperlink from p to q), then, in the graph model of the Web, there is an edge connecting p and q, directed from p to q.

The structure of the Web is important for supporting the process of information retrieval because structure is an embodiment of meaning and relationship. Existing research has tried to characterize the Web structure, to better understand the dynamics of the Web and to build models that capture these dynamics [39].

Link analysis has been also used in the computation of relevance measures for Web pages. Traditional information retrieval techniques that take into account only term characteristics for determining document relevance are not sufficient on the Web, due to the particularities of the Web. First, there is what has been called in the literature the "*abundance*" problem: for a topic of large breadth, there are a huge number of pages that could be returned as relevant (i.e. all the pages that contain the query keywords). However, many of

these pages are only marginally relevant to the topic of interest or may contain material of poor quality. Second, in many cases Web pages are *not sufficiently self-descriptive*. For instance, the *homepages* of important search engines, such as AltaVista [1] or Excite [16] do not contain the terms "search" and "engine". We can add to that the *synonymy* (a keyword might not appear in a document that is relevant, but rather its synonyms) and the *polysemy* problems (the same keyword might have different meanings in different contexts). All of the above make the application of traditional information retrieval techniques insufficient for the Web.

Analyzing the Web structure among Web pages helps address some of the above identified problems. This is because hyperlinks encode latent human judgment ([24], [31]), that is the creation of a hyperlink from page p to page q means that the author of page p has conferred to some extend *authority* to page q. Many important algorithms that are applied nowadays for information retrieval on the Web are based on the above statement. The HITS algorithm [24] categorizes Web pages in *authorities* (pages of high quality, authoritative on the topic of interest) and *hubs* (pages that link to many related authorities) and computes two measure of relevance for a Web page: an authority weight and a hub weight. PageRank [31] used by Google to rank the results of a query is another way of measuring the relevance of a Web page. Intuitively speaking, the more "important" pages point to a Web page, the more "important" this page becomes. PageRank uses the linkage structure of the Web to compute an "importance" score for each page on the Web. The pages returned by the search engine in response to a query are then ordered according to this score, i.e., their PageRanks, and the final ordering is presented to the user. Web structure has been also used to find out the "reputation" of Web pages [34], i.e. what are the topics on which a given Web page is known for. The algorithm explicitly incorporates document content of a given page and linked pages as a factor in the computation ranking of the given page.

Another important application of linkage analysis is the identification of communities on the Web. In [26], a method is proposed for inferring these communities from the topology of the Web. This is important for studying

the dynamics of the Web, from both social and historical perspectives.

In the "Related Work" chapter, we are going to discuss in more detail some of the important algorithms that leverage the structure of the Web in order to improve the retrieval of information on the Web.

Although a large body of work studies the use of linkage structure of the Web in order to improve the search for information, few existing papers in the literature address the problem of finding related pages on the Web ([15], [38]). Several search engines provide a "related pages" service (e.g., [20]), but the techniques used to retrieve related pages are typically not publicly documented.

Most of the algorithms for retrieving related pages on the Web are based on a technique descending from the field of bibliometrics, called *Co-citation* [37]. This technique counts the rate at which two documents are cited together in the same citation list. This rate is used as a measure of relatedness of two documents, i.e., the higher the rate, the more related they are. In the context of the Web, links inside Web pages can be interpreted as *citations*, and pages are the more related the more often their URLs appear together in link collections of other pages.

In 1997, Spertus [38] noticed that Co-citation could be used in the context of Web to find related Web pages. In [32], Co-citation is used as a similarity measure for clustering Web pages. The most consistent work that addresses the problem of finding related pages on the Web is the work of Dean and Henzinger [15]. They proposed in 1999 two algorithms for finding related pages on the World Wide Web. Both algorithms are using only the hyperlink structure of Web. The first one, called Companion, is based on HITS algorithm. We will review this algorithm in the "Related Work" chapter since is highly relevant for our work. The second algorithm is called Co-citation, and, as the name says, it is based on the co-citation technique. Co-citation algorithm will be presented in detail in the "Related Work" chapter and analyzed in the chapter 3 ,"Problems with Co-citation", since this algorithm is at the core of our work.

1.3 Our contribution

In this thesis, we have analyzed Co-citation [15] and identified a number of problems that frequently arise on the Web and may affect the precision of algorithms that use the link structure of the Web to find related pages:

1. Navigational links
2. Near-duplicate pages
3. "Unfocused" pages, which are, intuitively speaking, pages that contain a non-related collection of links.

Some proposals to deal with these problems have been described in the literature. Techniques are proposed to eliminate navigational links [14] and near-duplicate pages [4]. To deal with "unfocused" pages, a topic-independent notion of a "pagelet" has been formalized in [2]. Our experimental evaluation shows that, in the best case, only slight improvements in the results are obtained when using these "preprocessing " methods (it is possible that the quality of the results of the original Co-citation decreases after applying these techniques).

We propose a new notion of "focus" for a collection of links, which measures the degree of agreement on a topic between the corresponding pages. The goal is that in order to produce better results when looking for related pages an algorithm has to give more focused collections of links a higher influence on the final ranking than less focused collections. We formalize the notion of "focus" in three ways: link-based, content-based, and hybrid. We embed these notions into a "focused" version of the Co-citation algorithm and show in an experimental evaluation that these focused versions of Co-citation outperform the unfocused version with respect to the precision of the results, regardless of the use of the preprocessing techniques.

Our content-based notion of focus is related to the notion proposed in [28], which was successfully applied for finding the reputation of Web pages.

We propose a "customized" search for related pages. Because the notion of "relatedness" is subjective, we have built an interactive application, where

we allow the user to specify his interests by choosing the keywords that he believes relevant for the topic of the query URL.

The rest of the thesis is organized as follows. Chapter 2 gives an overview of the work related to this thesis. Chapter 3 discusses problems with the original Co-citation algorithm. One suggestive example is presented step by step to illustrate our argument. Chapter 4 formalizes the notion of focus, which is intended to overcome the identified problems. We propose several definitions of focus, based on content and linkage structure. All approaches are experimentally evaluated in Chapter 5 and the results show that our focused version of Co-citation consistently outperforms the unfocused version. Finally, we present our conclusions and we give directions for future work.

Chapter 2

Related Work

2.1 WWW and Web Mining

The World Wide Web (WWW, Web) is a huge collection of hypertext and hypermedia documents on a diversity of topics and interests, hyperlinks and access information.

The Web exhibits a number of idiosyncrasies that make knowledge extraction and resource discovery on the Web a very challenging task:

- The Web is a huge, widely distributed network, growing at a fast rate. Researchers from the Online Computer Library Center have estimated for the year 2002 that the number of unique web sites is 8,712,000 as compared to 4,662,000 in 1999 [29].

- The Web is heterogeneous and lacks a unifying structure. The complexity and diversity of Web pages if far greater than that of any traditional document collection: Web pages could be either static (HTML pages) or dynamic (generated from underlying databases through technologies such as .NET);some Web pages are poor in textual content, consisting mainly of images or may contain many idioms (navigational panels, templates, or advertisement banners etc.).

- The Web is highly dynamic. Information on the Web is constantly changing: pages are created or removed; Web services update regularly their Web pages. Link information and access records are also updated frequently.

Since the advent of the Web, a novel area came into shape, Web mining, which could be generally described as the process of extracting patterns from Web data through content mining, structure mining and usage mining. When mining the Web, a number of general problems arise:

- The *"abundance" problem.* For a topic of large breadth, there is a huge number of pages that could be returned as relevant. However, many of these pages are only marginally relevant to the topic of interest or may contain material of poor quality. The challenge is to be able to retrieve exactly those high quality pages that are indeed relevant for a specific topic.

- *Limited* coverage of the Web. Many important resources are actually hidden in underlying databases and the access to information is achieved by dynamically generated Web pages. Security and privacy issue have to be taken into account when mining this kind of information.

- *Limited* query interface. Searching for resources on the Web is mainly keyword-oriented. Although useful in many cases, this kind of searching is not enough to satisfy users' information need. Some users commit mistakes when choosing the keywords, so that the results obtained will be only marginally relevant to the users' interests. The retrieval and ranking strategies used by search engines cause sometimes difficulties when trying to pick the "right" keywords for expressing a query. Some alternative search interfaces exist already: "related pages" services in function (Google [20], Teoma [40]) or resource directories (Yahoo! [42], Open Directory [30]).

- *Limited* customization to individual users. The results of a given query to a search engine are identical, independent of the user. In order to improve the quality of the retrieved results, more information about the context in which the query has been submitted or the personal interests of the user has to be taken into account [27].

From a high-level perspective, the Web Mining field can be divided into three sub-areas:

- Web Content Mining: the process of extracting knowledge from the actual content of the Web pages.
- Web Structure Mining: the process of extracting knowledge from the hyperlink information existing on the Web. Often, Web content mining and Web structure mining are blended together.
- Web Usage Mining: the process of extracting knowledge from Web log records in order to gain deeper insights into the Web dynamics, which in turn could be used for many useful applications: identify potential customers for e-commerce, customization for individual users, etc.

Recent research in the Web mining field came to acknowledge that the hyperlink structure of the Web could be a very valuable tool for locating information ([15], [38], [32], [24], [10], [9], [13], [5], [12], [11], [31], [18]). The focus of this thesis is on Web Structure mining, or *linkage analysis*, as it is often referred in the literature. In the following section, we will review some of the most important applications that leverage on linkage analysis.

2.2 Web Structure Mining

Web structure mining or *linkage analysis* has its roots in the field of *bibliometrics*, which can be defined as the study of written documents and their citation structure. Bibliometrics is important to the Web studies because the Web could be regarded as a huge publication network in which the "prestige" of individual documents (Web pages) depends upon community recognition, which is often certified by direct hyperlinking.

Citation indexes, i.e. indexes that describe explicitly the linkage of citations between papers, have been used for searching and management of the information for a very long time. The oldest major citation index is, apparently, Shepard's Citations [36], developed in 1873 for indexing law-related

material (other authors suggests that the earliest form of citation indexing exists in the 14^{th}-century Hebrew texts [41]). The 20^{th} century brought an explosion of scientific literature so that new tools were needed to manage all this new information. E. Garfield developed *Science Citation Index (SCI)* [17] as an automated tool for document searching. SCI has proved to be a simple, but effective tool: it takes into account the authors' citations instead of an indexer's subjective judgments, so that many of the problems associated with the term and title-based analyses are avoided.

One of the problems addressed within the field of bibliometrics was to measure the similarity between a pair of documents. Two basic similarity functions on documents emerged: *bibliographic coupling* (Kessler [23]) and *co-citation* (Small [37])). Given two documents p and q, bibliographic coupling counts the number of documents cited by both p and q, where co-citation counts the number of documents that cite both p and q. Both these measures have been applied on the Web in the context of diverse applications, e.g., clustering of Web pages or the retrieval of related pages.

Bibliographic coupling seems to be unsuitable for the Web, because is susceptible to spamming. For example, a person could cite (i.e., link to) in his or her *homepage* all the Web pages cited also by Yahoo!'s *homepage*, so that bibliographic coupling will deem the person's *homepage* and Yahoo!'s *homepage* as similar, which will be obviously misleading. This kind of spamming is not possible within collections of scientific papers: a scientific paper does not cite other papers arbitrarily; it cites the papers that are to some degree relevant to it.

Co-citation is based on the opinions of independent authors and it is less susceptible to spam, so that it is more suitable for Web linkage analyses. Theoretically, if one would want to mislead co-citation, i.e., to determine co-citation to consider two pages p and q similar, although they are not, one would have to create a large enough number of Web pages that constantly cite p and q together. These fake pages should not be identical and should not reside on the same Web site, which is more difficult to achieve. The applicability of co-citation on the Web will be detailed in the section 2.3.2 of this chapter.

Web structure mining or linkage analysis has been used for different purposes: 1) to study the Web structure, 2) to compute the relevance of Web pages or 3) to identify communities on the Web. We will review each of the above application areas and survey some of the most important algorithms and results.

2.2.1 Studies of the Web structure

The study of the Web graph reveals valuable information that could be used not only to improve searching, crawling and community discovery on the Web, but also to study the sociological phenomena that characterize its evolution. Empirical observations have been drawn for a number of measurement experiments on the Web:

- The in-degree and out-degree of nodes in the Web graph follow a power law distribution [25].
- The Web exhibits a "small world" phenomenon. In a graph with a "small world" topology, nodes are highly clustered, yet the minimum distance between any two nodes in the graph is short. Studies of the Web ([8], [25]) have shown that the Web contains a giant strongly connected component (SCC), i.e., a set of Web pages such that for all pairs of pages (u, v) in this set, there exists a directed path from u to v. The average connected distance is defined as the average length of the path over all pairs for which the length is finite. If we consider the Web undirected, the average connected distance was measured to be 6.83 [25].

Other authors have imagined new theoretical models for the Web. Kleinberg et al. [25] proposed a new random graph model that reflected the process of content-creation on the Web by applying link-copying operations in the construction process. Pirolli et al. [33] discussed other two graph structures for the Web, one representing the similarity between the textual content of pages and another representing the usage patterns of the users ("the clicks-through"). Other perspectives try to build models that capture the complex dynamics of the Web [39].

The most often encountered model of the Web is the one in which the Web is seen as a directed graph, where nodes represent Web pages and an edge directed from node p to node q represents the hyperlink from page p to page q. Other similar approaches model the Web at different levels of granularity. For instance, the Web could be modelled as a directed graph, where the nodes represent Web sites, instead of Web pages, and a directed edge is drawn from site P to site Q if at least one of the Web pages existing within P links to at least one of the Web pages existing within Q.

Our understanding of both structure and dynamics of the Web can and should be used in solving the problem of information searching on the Web, but not necessarily as a stand-alone tool, but combined with contextual information, such as text or usage information.

2.2.2 Relevance of Web pages

Link analysis has been also used in the computation of relevance measures for Web pages. Techniques that take into account only term characteristics for determining document relevance are not suitable for the Web because of specific problems that appear on the Web. On the Web, there might be a large number of pages matching the query keywords, so that the challenge is to *distill* those high-quality pages that are indeed relevant to the query (the "*abundance*" problem). Some pages are poor in textual content: they contain mostly images and/or navigational-related information. Other pages contain "confusing" terms, such as misspelled words or words irrelevant for the topic of the page. Many pages are not sufficiently self-descriptive. For instance, the *homepages* of important newspapers, such as Washington Post (*http://www.washingtonpost.com/*) or Time (*http://www.time.com/time/*) do not contain the term "newspaper", but they rather have a varying content, according to the news of the day. To that we can add standard problems from the information retrieval field, such as the synonymy and the polysemy problem.

Existing research has tried to address some of the above identified problems by exploiting the linkage structure between Web pages. The majority of the

algorithms used today for resource discovery and knowledge extraction on the Web are based on the assumption that hyperlinks enclose meaningful relationships between pages. Some of these algorithms take into consideration only the existing linkage structure on the Web; however, there are several variants that combine the linkage information with the textual content of Web pages in order to improve the precision of the retrieved results.

HITS and its variants

Kleinberg developed the HITS algorithm (Hyperlinked-Induced Topic Search) [24] in order to address the "abundance" problem on the Web: given a search topic (in the form of a query), output high-quality pages that are highly relevant to the topic query. Such pages are called *authoritative*.

The HITS algorithm exploits only the linkage structure of the Web. The "importance" or how "authoritative" a Web page is with respect to the query topic should be assessed based on the collective endorsement of that page by diverse authors on the Web. The basic intuition is that the authority of the page is proportional to the number of its in-links: the more in-links a page has, the more "important" it is. However, the Web exhibits a number of particularities that make the above intuition invalid. First, not every hyperlink represents the endorsement that we seek. There are links created for other purposes such as navigational or paid advertisements links. Such links are called "nepotistic" (see section 2.5). Second, because of competitive interests, authoritative pages on some topic will rarely point to each other. For example, *Pepsi* might prefer not to endorse the competition's page, *Coca Cola*. Third, not even the textual content can help, because, as discussed above, many authoritative pages are not self-descriptive.

These unique features of the Web linkage structure led to the consideration of another important category of Web pages, the *hubs*. A *hub* is a Web page that provides collection of links to authorities. Hubs might not be prominent themselves or they might have few in-links. Examples of hubs are lists of recommended links on individual *homepages* or professionally assembled resource lists on commercial sites.

The input to the HITS algorithm is a search query (given as a set of keywords), which is sent to a search engine (in the original paper, AltaVista [1] was used). The top-most pages (in the paper 200 pages) returned by the search engine form the *root* set. Because many of the pages from the root set are presumably relevant to the search topic, some of them should contain links to the authorities on the topic. The *base* set is obtained by expanding the *root* set with all the pages linked by some page in the root-set and all of the pages that link to a page in the root set, up to a designated size cut-off (in the range 1000 to 5000). Hyperlinks between pages on the same host (where the *host* is assumed to be determined from the URL string) are not considered, because they are assumed to be by the same author and hence not indicators of value. The base set together with the existing hyperlinks between its pages is called the *neighborhood* graph. Let the neighborhood graph be G = (V,E).

On the resulting neighborhood graph, a weight-propagation scheme is proposed. The main idea is that hubs and authorities have a mutually reinforcing relationship. A page that points to many other pages is a good *hub*, and a page that many other pages point to is a good *authority*. It follows that a page that points to many good authorities is even a better hub, and a page pointed to by many good hubs is even a better authority. Every page (node) $v \in$ V receives two scores: a *hub score h(v)* and an *authority score a(v)*, initialized to any positive number. The algorithm repeatedly updates hub and authority scores, a and h, as follows:

$$a(v) = \sum_{(u,v)\in E} h(u) \tag{2.1}$$

$$h(v) = \sum_{(v,u)\in E} a(u) \tag{2.2}$$

After each iteration, the score vectors a and h are normalized, i.e.

$$|h|_1 = \sum_v h(v) = |a|_1 = \sum_v a(v) = 1.$$

If A is the adjacency matrix for the neighborhood graph G ($\forall$ i, j $\in$ V, A [i, j] = 1, if (i, j) $\in$ E; A [i, j] = 0, otherwise), then the formulas (2.1)

and (2.2) could be expressed as

$$a = A^T h \tag{2.3}$$

$$h = Aa \tag{2.4}$$

where $|\mathrm{h}|_1 = |\mathrm{a}|_1 = 1$ and A^T is the transpose matrix of A (the matrix obtained by interchanging the rows and columns of A). Based on the power iterations method [19], it is shown that the algorithm converges and a will be the principal eigenvector of $A^T A$, and h, the principal eigenvector of AA^T. Pages with high authority scores are expected to have relevant content and pages with high hub scores are expected to contain links to relevant content pages.

Finally, the HITS algorithm outputs the top-most authorities and the top-most hubs.

The Clever system ([12], [11]) has been developed at IBM as an attempt to use the HITS algorithm as a foundation for a search engine. However, a number of problems have been shown to exist with the HITS algorithm.

Bharat and Henzinger [5] discovered what they called *mutually reinforcing relationship between hosts.* In many trials with HITS, they found a pair of distinct hosts so that a set of documents on one host pointed to a single document on the second host, which increased the hub scores of the documents on the first host and the authority score of the document on the second host, or, the reverse case, when a single document on the first host pointed to a set of documents on the second host, which caused a similar problem. Assuming that the set of documents on each host is authored by a single person/organization, mutually reinforcing relationship between hosts give excessive weight to the opinion of a single person. They proposed an edge-weighting scheme in order to make sure that each distinct host is worth one unit of voting power. Specifically, if there are k edges from pages on a first host to a single page on a second host, each edge will receive an *authority weight* of $1/k$. This weight will be used in the computation of the authority score of the document on the second host. If there are l edges from a single page on a first host to a set of pages on a second host, each edge will receive a *hub weight* of $1/l$, which will

be used in the computation of the hub score of the document on the first host. If an edge connects nodes on the same host, the weight of the edge will be 0. By assigning fractional weights to the edges, the algorithm makes sure that all the documents on a single host have the same influence on the document they are connected to, as a single document would. This was one of the major improvements to the original approach.

It has been also noticed that HITS does not perform too well when the search topic is not sufficiently broad, so that not enough relevant pages will exist in the neighborhood graph G. In this case, broader topics will be represented by a denser sub-graph of hubs and authorities in G and HITS will return results relevant to the broader topic. This phenomenon is called *topic drift*. Sometimes, the broader topic is a natural generalization of the query topic. For instance, results for the query "movie awards" drifted to results pertaining to the more general domain of "movie companies". This is called *topic generalization*.

Different approaches were devised to deal with these problems. Clever system addressed these problems in two ways. First, if a fixed number of query terms are found within the anchor text and the surrounding context ("activation window") of a link, the edge corresponding to the link in the neighborhood graph is assigned a larger weight. The size of the "activation window" was tuned by trial-and-error. Second, at the origin of the *topic drift* problem is the presence of "mixed" hubs, which are pages that include a collection of links of which only a small subset was relevant to the query. This could cause authority scores to diffuse from relevant links to less relevant links. The proposed solution was to break "too long" hubs at prominent boundaries (such as <UL> or <HR>) into "pagelets", defined as contiguous sets of links.

In order to address the same problems, Bharat and Henzinger [5] computed a vector space representation of the documents in the base set and then pruned off pages that were "too far" from the vector space centroid in terms of cosine similarity measure. This technique is efficient for improving *precision*(i.e., the percentage of retrieved documents that are in fact relevant to the query topic), but it might reduce *recall*(i.e., the percentage of documents that are relevant

to the query and were, in fact, retrieved) since several mixed hubs are pruned. In the case of broad queries, losing a few hubs might not be a problem, but for narrow queries this technique might distort the results.

Page Rank

PageRank [31] is the algorithm used by the popular search engine Google [20] to rank its crawled Web pages. However, PageRank is not the only component used by Google to improve the quality of the retrieved results. Google uses several other features, such as the anchor text, or visual clues embedded within the Web pages (e.g., smaller fonts versus larger fonts). More details on the "anatomy" of Google are provided in [6].

Page et al. proposed a measure of the relative "prestige" of a Web page, called *PageRank*. The PageRank for a particular page is recursively defined as the weighted sum of the PageRanks of the pages that point to it. The rank of each page is propagated evenly to its outgoing links. The "random surfer" model captures this computation: a user "surfs" the Web by clicking on successive links at random and, from time to time, it is possible to jump to a random page with a given probability D (this probability corresponds to the "dampening factor" used in the PageRank's formula). The PageRank score of a page is the probability that the surfer visits this page when traversing the Web according to the above model.

PageRank scores are computed off-line for all the pages crawled by a Web crawler, as opposed to the authority and hub scores, which are computed dynamically in response to a given query. Another important difference between PageRank and HITS is that PageRank has no notion of hubs: pages only confer authority to each other. PageRank would not give a high rank to a page that points to many good authorities, but it is not pointed to by many pages. These hub pages are usually good collection of links that users find useful as overviews.

Page Reputation

Rafiei and Mendelzon [28] proposed a method for discovering the reputation of a given Web page in the context of some topic. The paper presents a search process where the input is the URL of a page and the output is a ranked set of topics on which the page has a reputation: e.g., if the input URL is *www.gamelan.com*, then a possible output is "Java".

The algorithm explicitly incorporates textual information with pure link analysis. The paper proposes two measures that relate a page p and a topic t: the *penetration* of page p on topic t, which is defined as the fraction of pages on topic t that point to page p and the *focus* of page p on topic t, defined as the fraction of pages pointing to p that are on topic t. For the purpose of this algorithm, a page p is *on topic* t simply when it contains the term of phrase t. This notion of focus of a Web page p on the topic t is similar in intuition with our notion of "focus", introduced in chapter 4, "Focused Co-citation". However, our notion of focus is designed in a different setting, having a different purpose, and we propose several, more complex schemes for its computation.

2.2.3 Web communities

Communities on the Web are simply defined as sub-graphs of Web pages that have more links among themselves than to pages outside the graph and correspond to groups of users that share a common interest [26]. Examples of Web communities are newsgroups, webrings or resource collections in directories such as Yahoo [42].

Several approaches have been proposed for finding such communities on the Web. The method proposed by Kumar et al. [26] "trawls" the Web for graph structures that are indicative signatures of the communities. They show that usually a community contains a *bipartite core* (i.e., a complete bipartite graph) and that co-citation is an early indication of emerging communities. Gibson et al. proposes a method for identifying Web communities based on identifying hubs and authorities on the Web [18].

Identifying and distinguishing between Web communities could be used to the benefit of search engines (i.e., focus their search on narrow but topically related subsets of the Web) and portals (i.e., target the advertisement at a very precise level), as well as to the studies concerned with the sociology of the Web.

2.3 "Related pages" algorithms

Although a large number of schemes that exploit the structure of the Web have been proposed, few existing papers address the problem of identifying related pages on the World Wide Web.

The majority of the algorithms that aim to find related pages are based on a technique descending from the field of bibliometrics, called *co-citation*. Small [37] developed in 1973 co-citation analysis as a method for assessing the common intellectual interest between a pair of documents. He noticed that the frequency at which two documents are cited together in citation lists is an indication of the topical relationship between the two documents: the higher the frequency, the more related the documents are.

Spertus [38] emphasizes the need for exploring not just the text within the Web pages, but also the linkage structure of the Web, in order to build effective Web information retrieval tools. A system, called *ParaSite*, is presented in the paper [38], that attempts to mine structural information on the Web by leveraging the many forms of links that exist on the Web (e.g., hyperlinks within a site could be categorized as *upward*, *downward* or *crosswise* with respect to the file hierarchy; hyperlinks to other sites are referred to as *outward*). The paper discusses the variety of link information existing on the Web, the differences between the Web and conventional hypertext and gives potential applications of link based analyses. Such an application is the discovery of related Web pages. Spertus notices that co-citation could be used in the context of the Web to find related pages. That is, if page A points to both pages B and C, then B and C might be related.

Pitkow and Pirolli [32] uses co-citation as a similarity measure for cluster-

ing Web pages in order to automatically categorize and aggregate hypertext documents. They notice that the reasons that motivated the development of citation and co-citation analysis are applicable also in the case of the World Wide Web. Citations in scholarly papers link documents to related documents. Similarly, on the Web, hyperlinks (when not used randomly) provide semantic linkage between the Web pages. Using only co-citation analysis as a similarity measure might reveal interesting topological patterns that in turn correspond to the semantic structure of communities of knowledge.

A technique for finding related *items* (such as Web pages, or compact disks) to a given item K is *collaborative filtering* [35]: the items related to K are those items that have been liked/recommended by other users who also liked K. Recommendations to a user are based on the preferences of other users that have similar profiles. The assumption is that items that have been considered likeable/similar by one user are going to be considered likeable/similar by another user with a similar profile. In the case of the Web, if a person links to pages P and Q, we might expect that people who like P may also like Q, especially if the links to P and Q are close to each other in the referencing page ([38]).

The most consistent work that we are aware of is the paper of Dean and Henzinger "Finding related pages on the World Wide Web" (1999) [15] that proposes two algorithms for finding related pages: Companion and Co-citation. Both algorithms use only the hyperlink structure of the Web to identify related Web pages. They neither exploit the textual content of Web pages, nor do they examine patterns of how users tend to navigate among pages. The only information being used is the order in which the links appear within a web page.

In the next two sections, we will present in detail the Companion and Co-citation algorithm, since they are highly relevant for this thesis.

2.3.1 Companion algorithm

In this section, we present the *Companion* algorithm [15] that identifies related Web pages to a given page by taking into account only the linkage structure

existing amongst Web pages and the order of the links within a Web page.

In order to present the Companion algorithm, we need to introduce some notation. The Web is modeled as a directed graph, $G = (V, E)$, where the set of nodes V corresponds to Web pages and the edges E corresponds to the existing links between pages: an edge in the graph G between node p and node q corresponds to a link from page p to page q, in this direction. A page p is called a *parent* of page q if there is a link in page p pointing to page q. In this case, we call q a *child* of page p.

The Companion algorithm is based on the HITS algorithm [24]. It takes as input a query URL u and outputs a set of related pages. It consists of the following steps:

Input: a Web page given by its URL u.

1. Build a *vicinity graph* for u.
2. Contract duplicates and near-duplicates in this graph.
3. Compute edge weights based on host-to-host connections.
4. Compute a hub score and an authority score for each node in the graph and return the top ranked authority nodes.

Output: return as related pages the pages with the 10 highest authority scores.

Figure 2.1: Companion algorithm

These steps are described in more detail bellow.

Step1: building the vicinity graph The *vicinity graph* consists of the query URL u, at most B (back) parents of u, at most F (forward) children of u, and for each selected parent (child) of u, up to BF (FB) of its children (parents) different from u. Selecting BF (FB) children (parents) for each parent (child) of u exploits the order of links within a Web page. Specifically, if a parent p of u has more than $BF+1$ children, then the algorithm selects up to $BF/2$ children pointed to by the $BF/2$ links on p immediately preceding the link to u and up to $BF/2$ children pointed to by the $BF/2$ links on p

immediately succeeding the link to u (ignoring duplicate links). If a parent p of u has less than BF children, the algorithm selects all its children. Analogously, for every child page c of the query URL u the algorithm selects up to FB parents "surrounding" (i.e., preceding or succeeding the link to u, except u itself. Figure 2.2 illustrates the construction of the *vicinity graph*:

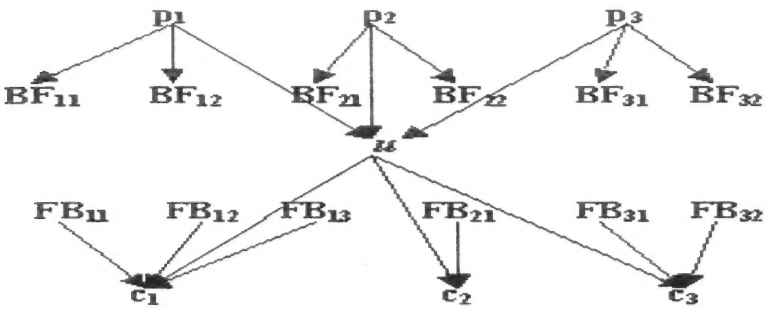


Figure 2.2: Companion: building the vicinity graph

A stop-list of 21 URLs (e.g., *www.yahoo.com*, *www.microsoft.com/ie/download.html*) is used to exclude "popular" pages from the vicinity graph only when the query URL is not a "popular" URL itself (if the query URL is a "popular" page, then the "popular" pages are allowed in the vicinity graph because they might be in fact related). "Popular" pages are pages that are unrelated to most queries and have very high in-degree (the *in-degree* of a given Web page is defined as the number of pages that point to the given page).

When building the *vicinity graph*, edges between nodes on the same host are omitted. The paper mentions briefly that the host can be determined from the URL string, but the heuristic used is not explained.

The selection of BF (FB) children (parents) when building the vicinity graph is based on the assumption that links to pages on a similar topic "tend to be clustered together, while links that are farther apart on a page are less likely to be on the same topic" [15].

Step2: contracting duplicates and near-duplicates After building the vicinity graph, the duplicate and "near-duplicate" nodes are contracted. Two

nodes are near-duplicates if (a) they each have at least 10 links and (b) at least 95% of their links are in common. When two near-duplicate nodes are found, they are merged into a single node, whose links are the disjoint union of the links of the two near-duplicates.

The authors of the paper empirically noticed that allowing "near-duplicate" nodes in the vicinity graph greatly distorts the precision of the Companion algorithm's results.

Step3: assigning edge weights The edge-weighting scheme assigns weights to the existing edges in the vicinity graph, as described in the previous work on topic distillation [5] to address the problem of *mutually reinforcing relationship between hosts* (see 2.2.2).

Step4: computing hub and authority scores Hubs and authorities scores are computed for each node in the vicinity graph using a modified version of the HITS algorithm, which takes into consideration the previously computed edge weights.

The documents with high authority scores are expected to have relevant *content,* and the documents with high hub scores are expected to contain *links* to pages with relevant content. The algorithm returns as related pages the top 10 highest authority scores.

Kleinberg suggested in [24] that HITS could be used for finding related pages on the Web. However, Companion builds a slightly different neighborhood graph and brings a number of improvements, such as exploiting the order of links within a Web page, elimination of near-duplicate pages from the neighborhood graph and the edge-weighting scheme (which did not exist in the original HITS).

2.3.2 Co-citation algorithm

In this section, we present the Co-citation algorithm that is at the core of our work. Similarly to the Companion algorithm, Co-citation exploits only the linkage structure existing amongst Web pages and the order of the links

within a Web page in order to identify related pages.

We need to introduce some notation in order to present the Co-citation algorithm. A *sibling* of a page p is another page q, so that p and q have at least one parent in common. We will refer to two pages, p and q, as being *co-cited* if and only if they have at least one common parent. The *degree of co-citation* for a pair of pages is defined as the fraction of parents that the two pages have in common out of the total number of parents.

The Co-citation algorithm as described in [15], takes as input a Web page, given by its URL u and outputs a set of related web pages. As its name says, this algorithm is based on the co-citation technique: the higher the frequency two documents are cited together in the same citation list, the more likely they are similar. In the case of the Web, the algorithm is examining the siblings of the query URL u. The higher the degree of co-citation between a sibling and the query URL, the more related is this sibling to the query URL. The Co-citation algorithm consists of the following steps:

Input: a Web page given by its URL u.

1. Build a *vicinity graph* for u.
2. For each of the siblings of the query URL u, compute the degree of co-citation with u.

Output: return as related pages the pages with the 10 highest degrees of co-citation.

Figure 2.3: Co-citation algorithm

A detailed description of these steps follows:

Step1: building the vicinity graph The vicinity graph consists of the query URL u, at most B(back) parents of u, chosen at randomly from the set of parents of u, and for each of the chosen parents, p, at most BF (back-forward) outgoing links (children), so that these links surround the link to u. The selection of the BF outgoing links for each of the considered parents is identical to the selection of children explained in the Companion algorithm:

for a parent p, the algorithm selects up to $BF/2$ links that appear on the page p immediately before the link to u and up to $BF/2$ links that appear immediately after the link to u. This is the part of the algorithm that exploits the order of links within a Web page and it is based on the assumption that links that appear close together in the same Web page are likely to be on the same topic (related).

Figure 2.4 summarizes the construction of the vicinity graph:

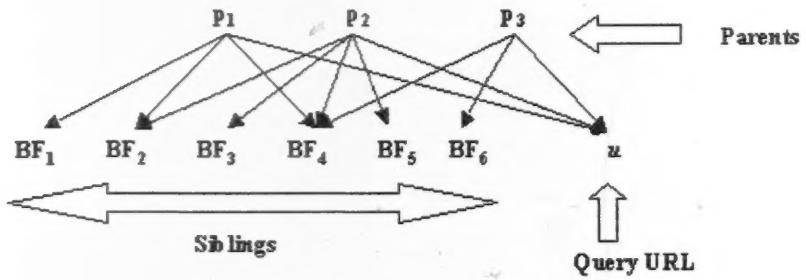


Figure 2.4: Co-citation: building the vicinity graph

Step2: computing degrees of co-citation The set of children selected for each of the parents in the vicinity graph forms the set of siblings. For instance, in figure 2.4, each of the Web pages BF_1, BF_2, BF_3, BF_4, BF_5, BF_6 is a sibling of the query URL, u. For each one of these siblings, the algorithm computes the degree of co-citation with u. The degree of co-citation for a sibling and the query URL u is computed as the number of parents that they have in common divided by the total number of parents in the vicinity graph:

$$Degree\_of\_co-citation(BF_i, u) = \frac{no\_common\_parents}{no\_all\_parents}$$

In the above picture, for the sibling BF_2 and u, the computed degree of co-citation is 2/3, because BF_2 has 2 parents in common with u (i.e., p_1 and p_2) and the total number of parents in the vicinity graph is 3 (p_1, p_2 and p_3). Similarly,

$$Degree\_of\_co-citation(BF_1, u) = 1/3$$
$$Degree\_of\_co-citation(BF_3, u) = 1/3$$
$$Degree\_of\_co-citation(BF_5, u) = 1/3$$
$$Degree\_of\_co-citation(BF_6, u) = 1/3$$
$$Degree\_of\_co-citation(BF_4, u) = 3/3$$

The siblings are ranked according to the computed degrees of co-citation. The algorithm outputs the top 10 siblings with the highest degrees of co-citation as related pages to the query URL u. In the case of our example, BF_4 is ranked first, BF_2 second, followed by the rest of the siblings that have the same rank.

As the authors of the algorithm notice, when there is an insufficient level of co-citation with u, i.e. when the link structure around the starting URL is sparse, many unrelated siblings may be deemed as related. This is why, when there are not at least 15 nodes, which are co-cited with u at least twice, the last path element is removed from u and the algorithm is restarted with the new URL u'(e.g, if u = a.com/X/Y/Z, then u' = a.com/X/Y).

2.4 Work related to "pagelets"

Related to our thesis is also the work that has been done regarding the notion of a "pagelet". The intuition is that links that are close to each other within a Web page tend to point to pages on the same topic. This intuition has been exploited by some of the existing work as follows.

Chakrabarti et *al.* [10] uses the links and their order within a Web page for categorization of the Web pages. They show that the links that appear near a given link in the page order often point to pages that are on the same topic.

The name "pagelet" first appeared in [11] and a "pagelet" was defined as a contiguous set of links. The majority of work developed around the HITS algorithm explores to some extent the notion of a pagelet in an attempt to break the hub pages into more topical-cohesive units. However, the "pagelet" was never defined formally, its definition remained as vague as a "contiguous" set of links.

Chakrabarti [9] and Chakrabarti et al. [13] propose (2001) two algorithms for partitioning "mixed" hubs into pagelets in the context of the HITS/Clever algorithm. The first algorithm uses only the DOM (Document Object Model) representation of a Web page to split the hubs into pagelets, while the second algorithm takes into account also textual content in order to distinguish which pagelets are more topically cohesive. These algorithms are embedded within the HITS settings, are computationally expensive and partitioning of Web pages into pagelets is dependent on the given query.

In 2002, Bar-Yossef and Rajagopalan [2] formalize the notion of a "pagelet" and provide a semantic and syntactic definition, as well as an algorithm for extracting the pagelets from a Web page. Semantically, a pagelet is defined as a region of a Web page that

1. Has a single, well-defined topic or functionality; and
2. Is not nested within another region that has exactly the same topic or functionality.

A Web page is composed of one or more pagelets, corresponding to the multiple topics and functionalities encompassed within that page.

The paper provides also a syntactic definition that attempts to capture the semantic of a pagelet. The heuristic proposed constructs the HTML tree for a Web page and takes into consideration only certain elements of the HTML tree, such as tables, paragraphs, headings, list, etc., in order to decompose a Web page into pagelets. Such HTML elements might be useful clues when trying to extract the pagelets from a Web page: often, the creators of Web pages use such HTML mark-up tags to separate distinct regions. Syntactically, a pagelet is defined as an HTML element in the parse tree of a page p, so that:

1. None of its children in the HTML tree contains more than k hyperlinks; and
2. None of its ancestor elements in the HTML tree is a pagelet.

If an HTML element contains at least k hyperlinks, then it probably represents a distinct topic/functionality; otherwise, it is more likely to be topically integrated in its parent. Figure 2.5 illustrates the algorithm proposed for the extraction of pagelets:

```
Partition(p){
    T_p := HTML parse tree of p
    Queue := root of T_p
    while (Queue is not empty){
        v := top element in Queue
        if (v has a child with at least k links)
            push all the children of v to Queue
        else
            declare v as a pagelet
    }
}
```

Figure 2.5: Pagelet extraction algorithm

In our preprocessing phase, we adopt this definition and use the above algorithm to extract pagelets (see chapter 3).

2.5 Work related to "nepotistic" links

As many authors noticed ([24], [5], [31]), the creation of a link from page p to page q represents the endorsement of page q by the author of page p. However, on the web today there are many links that violate this assumption. Algorithms that exploit the linkage structure of the Web will be affected by the presence of such links that are present for reasons other than merit. Davison [14] calls these links "nepotistic". The paper uses a C4.5 classification algorithm on a large number of page attributes, trained on manually labeled pages, in order to assess the potential for automatic recognition of nepotistic links. Bar-Yossef and Rajagopalan discuss the same issue in [2] and provide instances of nepotistic links such as navigational, download, advertisement, agreement-exchange links, and even links introduced deliberately to mislead search engines.

Co-citation algorithm leverages on the linkage structure of the Web, so that it may face the problem of certain nepotistic links. We employ simple, but effective heuristics for deletion of the navigational links from our data set. The details are provided in chapter 3.

2.6 Work related to"near-duplicate" Web pages

Several papers addressed the phenomenon of "mirroring" on the Web, i.e., the systematic replication of content over pairs of hosts. Mirroring is understood mainly at the host level ([4], [3]), although it can manifest itself at the page level too ([15], [7]). Existing algorithms operate on the basis of URL strings, linkage data and content analysis. A comparison of several algorithms for identifying mirrored hosts on the Web is provided in [4].

Identification of mirrored hosts across the Web is important because of the following reasons. First, there are many algorithms that exploit the explicit linkage between Web pages. Mirrors hosts (documents) could greatly distort the results of such algorithms because they perturb the graph model. Second, search engines may benefit from identifying mirrored hosts: storing and returning duplicate documents in large amounts can be avoided. Third, proxies and other Web services may use mirror sites as an alternative to compensate for various failures and thus achieve improvements in performance.

Co-citation algorithm may face the problem of "near-duplicate" pages in the set of selected parents, as detailed in the next chapter. We employ the same heuristic for determining such near-duplicates as the one proved to be effective in the case of the Companion algorithm [15].

2.7 Work related to evaluation strategies

The different variations of Co-citation that we propose in this thesis produce, given a query URL, a ranked listing of related URLs (pages). An important step that needs to be performed after running the algorithms is the evaluation of the retrieved results, i.e. we need to assess if the results are indeed related to the query URL.

Because relatedness or similarity is subjective and difficult to measure, the results of "similarity-search" algorithms (such as Co-citation, Companion, etc.) are usually evaluated by conducting user studies. Such a user study has been performed for evaluating the results of Companion and Co-citation [15]. The users are requested to score each retrieved result. Some user studies have used a binary scoring scheme (a result is scored 1 only if the user perceives it as relevant to the query URL, otherwise 0 [15]), while others may allow a finer-grained scale to measure degrees of relatedness.

However, these user studies might be expensive in both time and resources, and they are not suitable for the cases where a large number of runs of an algorithm need to be evaluated. It would be better to have an automatic, reliable evaluation strategy that allows easily the evaluation of a large number of experiments.

Several techniques for automatic evaluation of the results of similarity-search algorithms have been proposed in the literature. Some authors have tried to compute a "coarse", domain-specific, similarity measure that, although far from being definitive or exhaustive, does serve to illustrate important aspects of the proposed algorithms ([9], [13], [22]).

Another evaluation strategy is to use a form of "ground truth" for relatedness and to evaluate the results with respect to it. Web hierarchies have the potential to act as such "ground-truth" forms. Haveliwala et al. [21] introduce a technique for automatically evaluating diverse strategies for similarity search on the Web. They use Open Directory [30] as a form of "ground-truth". They report in the paper that the best similarity search on the Web is the one that represents documents by considering anchor text and content (Jaccard coefficient was used as a distance function between documents). One of our notions of focus is formalized in a consistent way with their findings. We propose an evaluation methodology based on the Open Directory as a form of "ground truth". The details and a discussion on the use of Open Directory as a "ground truth" for measuring relatedness is provided in chapter 5.

Chapter 3

Problems with Co-citation

We have implemented the Co-citation algorithm as described in [15] and in the "Related Work" chapter and we have identified a number of problems that may affect the quality of the results of the algorithm.

We will present each problem and a potential solution using one suggestive example to illustrate our discussion. The example is the URL for the *homepage* of the computer science department at the University of British Columbia (UBC): *www.cs.ubc.ca*. We expect as related pages other *homepages* of computer science departments (preferably in Canada). We chose this query URL out of many other possible examples because we have a good intuition of what is related and what is not related to it, so that this example is easily understandable.

In the first section of this chapter we will present the results obtained by the Co-citation algorithm on our running example. Then we will discuss each of the possible reasons for the poor performance of Co-citation, that is:

1. Navigational links
2. Near-duplicate pages
3. Pages with links related to multiple topics

We will discuss what techniques we have applied to deal with these problems. The sequence of these techniques represents what we call the "preprocessing stage". We will motivate the need for a new notion, the "focus" of a collection

of links that holds the potential to significantly improve the results of the Co-citation algorithm. In the end, we analyze the effect of the preprocessing stage on our experiments.

3.1 Original Co-citation

For the rest of this thesis, we will refer to the Co-citation algorithm, as discussed in [15] and presented in the previous chapter, as the *original* Co-citation. We will present only the top 10 results returned by original Co-citation and its further improvements.

For our experiments, we have collected at most $B = 1000$ parents of the query URL and for each of the selected parents, we have extracted at most $BF = 8$ links surrounding the link to the query URL (see section 2.3.2). We have run the original Co-citation algorithm for the query URL *www.cs.ubc.ca*. The results are presented in Table 3.1. We can clearly see that the results are not very convincing. All the results belong to the UBC domain. The first result is the *homepage* of UBC and all the other results are pages belonging to the computer science department of the same university. We deem the results returned by the original Co-citation for this query URL unrelated.

Table 3.1: Original Co-citation - results

URL	DESCRIPTION
www.ubc.ca	UBC homepage
www.cs.ubc.ca/research/index.html	UBC CS Research Areas
www.cs.ubc.ca/about/index.html	UBC CS About the Department
www.cs.ubc.ca/prospective/index.html	UBC CS Prospective Students
www.cs.ubc.ca/grads/index.html	UBC CS Current Graduate Students
www.cs.ubc.ca/ugrad/index.html	UBC CS Undergraduate Program
www.cs.ubc.ca/people/index.html	UBC CS People
www.cs.ubc.ca/labs/imager	UBC CS Imager Laboratory
www.cs.ubc.ca/labs/imager/th.html	Imager Computer Graphics Laboratory Theses and Major Essays
www.cs.ubc.ca/events/index.html	UBC CS Events and Seminars

Some of the results presented in Table 3.1 (e.g., *www.cs.ubc.ca/*

grads/index.html, *www.cs.ubc.ca/people/index.html*) look like links that usually belong to a navigational panel included in all (or many) pages of a specific domain, in this case, the UBC computer science department's domain. After a brief inspection of the parents of the query URL, we noticed that many of these parents are in fact from the UBC computer science department's domain and that some of the above results are part of the same navigational menu (see Figure 3.1), included in most of these parents.

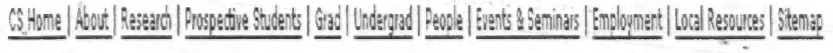

CS Home | About | Research | Prospective Students | Grad | Undergrad | People | Events & Seminars | Employment | Local Resources | Sitemap

Figure 3.1: Standard navigational panel

These observations made us consider the first phenomenon that may affect the performance of the Co-citation algorithm: the presence of navigational links within our link data set. We will detail this problem and our solution to it in the next section.

3.2 Navigational Links

As discussed in the "Related Work" chapter, the existence of "nepotistic" links on the Web ([14]), i.e., links between pages that are present for reasons other than merit, is likely to affect the precision of algorithms that leverage on the linkage structure of the Web. Instances of nepotistic links are navigational, download, advertisement, agreement-exchange links, and even links introduced deliberately to mislead search engines [2].

The Co-citation algorithm may face the problem of certain nepotistic links. In order to rank the siblings of the query URL, the algorithm counts for each sibling the number of parents that it has in common with the query URL. We try to prevent the influence of siblings that are related to the query URL due to other considerations than topical and also have a sufficiently large degree of co-citation with the query URL. These might be siblings that appear due to the presence of navigational links.

In our implementation, we aimed to eliminate those navigational links

- Between any parent and the query URL, and
- Between any parent and its selected children.

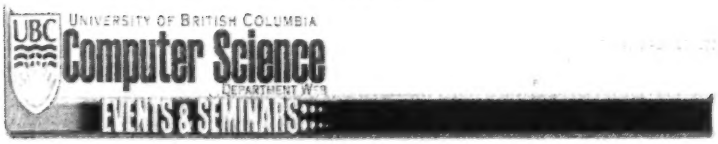
In order to identify navigational links, we used several heuristics, based on the URL strings and which differentiate between *homepages* and other pages. Details and examples are provided as follows.

We will ignore the protocol part ("http://","https://", etc.) when working with the URL strings.

We identify *homepages* by the presence of the character $\sim$ in the URL string. Examples of *homepages*, according to our heuristic, are pages such as *www.cs.ualberta.ca/~gabi*, *www.math.ubc.ca/~jf/pubs/index.html*, etc. We identify the *username* of a *homepage* as the URL sub-string starting after the first occurrence of the "$\sim$" character and ending before the first occurrence of the character "/" (if "/" does not occur, we take as the *username* the URL sub-string starting after the first occurrence of the "$\sim$" character and ending at the end of the URL string). For the two sample URLs above, the usernames are "*gabi*", respectively "*jf*".

Given the URL of a page, we define the *complete host name* as the string starting at the beginning of the URL string (ignoring "www." at the beginning if it is present) until the first occurrence of the character "/" (or the end of the URL string). For instance, if the URL string is *www.cs.ubc.ca/research/index.html*, then the complete host name of this URL will be the string "*cs.ubc.ca*".

If we have two complete host names given by their strings, s_1 and s_2, we say that s_1 is *more specific* than s_2, if and only if s_2 is a suffix of s_1. As an example, consider the following two hostnames: s_1 is "*ugrad.cs.ubc.ca*" and s_2 is "*cs.ubc.ca*". Hostname s_1 is more specific than hostname s_2, because s_2 is a suffix of s_1. A link from a more specific page s_1 to a more general page s_2 will be often a navigational link, for instance included in a standard menu that appears on many of the pages of the more specific domain and points to the larger domain.



Links and Statistics about Women in IT

- Undergraduate Programs at UBC Computer Science
  http://web.archive.org/web/20020620173923/http://www.cs.ubc.ca/ugrad/Program/Options/index.html
- How to Get into UBC for Prospective Students
  http://web.archive.org/web/20020620173923/http://www.ubc.ca/students/prospective/index.html
- Focus on Women in Computer Science (FoWCS)
  http://web.archive.org/web/20020620173923/http://www.cs.ubc.ca/~bam/FoWCS/FoWCS.html
- Links to Women's Organizations
  http://web.archive.org/web/20020620173923/http://taz.cs.ubc.ca/swift/links.html#women
- WOMEN IN HI-TECH FIELDS IN SCIENCE AND TECHNOLOGY IN BRITISH COLUMBIA (final report)
  click here to download the file in pdf
- WOMEN IN HIGH TECH FIELDS IN SCIENCE AND TECHNOLOGY IN BRITISH COLUMBIA FACT SHEET AND SUMMARY JULY 1999 (executive summary)
  click here to download the file in pdf
- Women and Computer Science
  click here to download the file in pdf
- High Tech Stats
- UBC stats on participation of women in science and engineering 2000
- UBC stats on participation of women in science and engineering 1997
- SFU stats on participation of women in science and engineering 1997
- Comparison of male/female population in all science units at UBC and across Canada

---

CS Home | About | Research | Prospective Students | Grad | Undergrad | People | Events & Seminars | Employment | Local Resources | Sitemap

Figure 3.2: Navigational links from a "more specific" to a "less specific" domain

For two URLs that are not *homepages*, we use the heuristic that a link between URL s_1 and URL s_2 is *navigational* if the host names are identical or if the hostname of s_1 is more specific than the hostname of s_2. In the last case, we say that s_1 is in relation "more specific – less specific" with s_2.

For instance, the URL s_1 = *www.cs.ubc.ca/ugrad/facilities/remote/news/* represents one of the parents of the query URL s_2 = *www.cs.ubc.ca*. The complete host name of s_1, "*cs.ubc.ca*", is identical to the complete host name of s_2, "*cs.ubc.ca*", so we consider the link from s_1 to s_2 navigational.

Figure 3.2 illustrates the parent URL s_1 = *taz.cs.ubc.ca/itweek/links.html* of the query URL s_2 = *www.cs.ubc.ca*. The complete host name of s_1, "taz.cs.ubc.ca", is *more specific* than the complete host name of s_2, "*cs.ubc.ca*", so that we consider the link from s_1 to s_2 navigational.

In figure 3.2, we can see that the link to the query URL belongs to a navigational panel inside the parent page. For this parent page, the set of siblings that Co-citation would consider are associated with the query URL due to navigational, not-topical reasons.

The last example illustrates the reason for which we decided to ignore the "www." at the beginning of the URL string. The "www." part of the URL string is common for many URLs, so it does not add any significant information. More, when we want to assess this type of navigational links, that we call "more specific – less specific", the "www." part entangles the computation: "*www.cs.ubc.ca*" is not a suffix of "*taz.cs.ubc.ca*", but "*cs.ubc.ca*" is such a suffix.

For *homepages*, we only do the same nepotistic link elimination only if the username for both URLs is the same. More precisely, if the complete host names of two URLs are identical or in relation "more specific – less specific" *and* the usernames are identical, we consider the link between them navigational.

As an example, consider the query URL s_2 = *www.cs.ualberta.ca/~stroulia*. A parent page such as s_1= *www.cs.ualberta.ca/~stroulia/661/Fall2001* will typically be a page belonging to the same person (the complete host names are identical, the usernames are also identical) that contains various information,

and, from navigational reasons, points also to the person's *homepage* (see Figure 3.3). We consider the link from s_1 to s_2 navigational.

CMPUT661 -- Software Architecture
Fall 2001
Department of Computing Science
University of Alberta

time	newsgroup	instructor
T R 11 00 - 12 20	*The class forum:* ualberta.courses.cmput.661 Evolving Class Schedule Class Projects Page What's New	Eleni Stroulia Dept. of Computing Science 105 Athabasca Hall University of Alberta Edmonton, AB, T6G 2E8, Canada Office: Athabasca 307 Phone: 1 780 492 3520 Fax: 1 780 492 1071 email: stroulia@cs.ualberta.ca

Figure 3.3: Homepages - navigational links

Let us consider another parent page: s_1= *www.cs.ualberta.ca/people/home-faculty.php*. Although the complete host names are identical, the usernames are different (in fact, s_1 is not even a *homepage* according to our heuristic, so the username for s_1 is the empty string); therefore we keep the link from s_1 to s_2, because it is not navigational. Actually, this particular parent may produce related siblings to the query URL, such as the *homepages* of other professors from the same department. This also explains why our heuristic differentiates between *homepages* and other pages: we need more information to identify navigational links in the case of *homepages*.

One last example concerns the case where the complete host names of s_1 (parent URL) and s_2 (query URL) are different, but the usernames are identical. In this case, we consider the link from s_1 to s_2 not navigational, because we could not know if there is the same person, having two *homepages* on two different hosts, or there are two different persons, incidentally having the same username.

We clean our link dataset by eliminating any parent that is linked to the query URL through a navigational link. These parents will produce siblings, which are related to the query URL typically because of navigational, not topical considerations (usually, navigational links are grouped together in a navigational menu). Often these siblings will be from the same site as the

query URL and when computing their ranks, they may get a high rank, because they may share a large number of parents with the query URL. Similarly, for each parent, we select only those children that are not pointed to through a navigational link. We call this operation "navigational links elimination".

We have applied "navigational links elimination" operation to the original Co-citation algorithm. The pseudo-code for the navigational links elimination applied to the original Co-citation algorithm is presented in Figure 3.4. The top 10 results are presented in Table 3.2.

Input: a Web page given by its URL u.

1. Collect up to B parents of u.
2. Eliminate parents that point through a navigational link to the query URL u.
3. Eliminate siblings that are pointed to through a navigational link by their parents.
4. Compute the degree of co-citation for the remaining siblings.

Output: return as related pages the pages with the 10 highest degrees of co-citation.

Figure 3.4: Navigational links elimination

We notice some improvement in the results. Three of the results are *homepages* of other departments from UBC. The other results are to some extent related to the computer science area: two companies that provide Internet related services, two links within computer science related areas (Cognitive Systems and Bioinformatics), and the *homepages* of two laboratories within the UBC computer science department. However, there is no *homepage* of other computer science department present in this list of results. The UBC *homepage* appears in the first position.

The initial set of parents crawled from Yahoo! [42] was comprised of 770 parents (URLs). Out of these parents, 536 parents were eliminated because they were linked through a navgigational link to the query URL. The large number of "nepotistic" parents explains the results obtained by original Co-

Table 3.2: Original Co-citation - Results after navigational links elimination

URL	DESCRIPTION
www.ubc.ca	UBC homepage
www.psych.ubc.ca	UBC Psychology Dept.
www.linguistics.ubc.ca	UBC Linguistics Dept.
www.philosophy.ubc.ca	UBC Philosophy Dept.
www.ec-o.com	Internet - related Company
www.bsdi.com	Internet Services
www.cs.ubc.ca/nest/lci	UBC CS Laboratory for Computational Intelligence
www.ams.ubc.ca/clubs/cogsys	UBC Cognitive Systems Society
www.cs.ubc.ca/nest/imager/imager.html	UBC CS Imager Laboratory
life.anu.edu.au:80	Bioinformatics Group, Australian National University

citation (Table 3.1), where all top 10 answers were internal pages of the computer science (CS) domain.

Even after the navigational links elimination phase, the results are far from what we deem as related to the query URL, i.e., *homepages* of other computer science departments. We performed a second investigation on our data and we have identified the second problem, which is discussed in the next section: the presence of near-duplicate pages within our set of parents.

3.3 Near-duplicate pages

The problem of "near-duplicate" pages has been addressed in the paper that originally proposed Companion and Co-citation [15], but only for the Companion algorithm. In the context of the Companion algorithm, two pages are "near-duplicates" if

1. They each have at least 10 children, and
2. At least 95% of their children are in common.

Companion merges two near-duplicate pages into a single page whose set of links is the disjoint union of the links of the two pages. This step is called

"near-duplicates contraction". It is not clear from [15] whether the authors performed any near-duplicates contraction for the Co-citation algorithm or not.

Co-citation algorithm might be affected by the presence of near-duplicate pages in the set of parents. We try to prevent the existence of siblings that are not related to the query URL; yet they have a sufficiently large degree of co-citation with the query URL so that they may be ranked high by Co-citation. Let us imagine the case when a parent page is replicated (mirrored) at several sites. Each of the siblings produced by this page will have an artificially increased rank in the Co-citation scheme. Instead of counting this parent once for each of its children, we count it several times. We want to make sure that in our set of parents each distinct parent gives one vote to its children.

We use the same heuristic for detection of near-duplicate parent pages as the one used by Companion, given by conditions (1) and (2) from above. This heuristic is based on the set of outgoing links of a Web page. The intuition is that pages with many links are likely to provide a better evidence of mirroring relationship than those with a small number of links. This fact was noticed by previous research [4]. Experiments performed in [4] showed that the minimum number of links that a page should have in order to be considered a candidate for near-duplicates contraction is 10 (larger values did not improve the performance too much). This explains the use of condition (1). Regarding condition (2), we want our heuristic to be flexible enough to allow detection of pages that are not necessarily identical, but they are still duplicated (this is because the sets of outgoing links may vary slightly across different sites, due to local customization). Since we want a high degree of agreement, we check if 95% of the children of two pages are in common.

As we will explain in the next section, we will extract a "pagelet" out of every parent page. The extracted pagelet is a collection of links (URLs), including the query URL. After this extraction step, our set of parents will consist of pagelets instead of whole pages.

In the case of the original Co-citation [15], for each parent page, we extract a set of *BF* links surrounding the link to the query URL (see section 2.3.2).

The set of extracted links represents the "pagelet" used by the original Co-citation. We differ in the way we extract the pagelet out of every parent page.

Given the fact that our set of parents will consist of pagelets, one obvious question is why we detect and contract parent *pages*, instead of detecting and contracting pagelets. The argument is that the fact that two pagelets are near-duplicates does not necessarily imply a mirroring relationship, i.e., does not necessarily mean that the pages from which the pagelets were extracted are also near-duplicates. Ideally, a pagelet should contain those links from the parent page that are related to the query URL. Let us imagine the case when for a parent page we extract a pagelet that contains a small number of related links to the query URL. Similarly, for another parent page, which is not a near-duplicate of the first parent page, our pagelet extraction algorithm produces the same (or very similar) set of related links to the query URL. This is exactly what Co-citation algorithm is looking for: distinct parents that express their vote on a set of children. Yet, the pagelets extracted would be deemed as near-duplicates and merged. In this way, we lower the rank that Co-citation would compute for the extracted set of related siblings. One issue involved in the detection of near-duplicate pagelets is what condition (1) should be. If we decide not to consider condition (1) at all, we will face the problem just explained above: we may affect the results of Co-citation by merging pagelets that do not represent any mirroring relationship. If we consider condition (1), it is very difficult to compute (if it can be computed at all) the minimum number of links that a pagelet should have in order to be considered a candidate for near-duplicates contraction. The extracted pagelet can vary in size (i.e., the number of its outgoing links) from very small pagelets (two, three links) to large pagelets (the number of links is of the order of hundreds). Intuitively, the number used in condition (1) should not be a small number because of the same considerations: if we detect pagelets with a large number of links that share a large fraction of those links, probably this is a good indication of a mirroring relationship at the page level. Setting the parameter for condition (a) requires a large number of systematic experiments and further investigations.

We formalize our near-duplicates contraction phase. We perform near-duplicates detection for the parent pages. Let us assume that A is a parent page. By $|A|$ we denote the number of outgoing links existing within the page A. By *MIN_LINKS* we understand the minimum number of links that a parent page should have in order to be considered a candidate for near-duplicates contraction. By *THRESHOLD_NDC* we understand the number of outgoing links that two parent pages should share in order to be considered in the near-duplicates contraction scheme.

We consider that two parent pages A and B are near-duplicates if and only if

$$|A| \geq MIN\_LINKS, |B| \geq MIN\_LINKS \tag{3.1}$$

$$\frac{|A \cap B|}{|A \cup B|} \geq THRESHOLD\_NDC, \tag{3.2}$$

where $|A \cup B| = |A| + |B| - |A \cap B|$, i.e., the common links between A and B are considered only once. Formula (3.1) and (3.2) represent the mathematical encoding of the condition (1), respectively (2). For our experiments, we used *MIN_LINKS* equals 10 and *THRESHOLD_NDC* equals 95%.

When we encounter two near-duplicate parent pages according to formulas (3.1) and (3.2), we will merge them into a single parent page, whose set of links is the disjoint union of the links of the two near-duplicate pages. We call this step "near-duplicates contraction".

We have applied both "navigational links elimination" and "near-duplicates contraction" operations (in this order) to the original Co-citation algorithm. The pseudo-code for the original Co-citation algorithm is presented in Figure 3.5. The top 10 results are presented in Table 3.3.

Our running example is not affected by the presence of near-duplicate parent pages. Out of 234 parents left after the navigational links elimination, only 2 parents were detected as near-duplicates and merged. Due to the small number of near-duplicate parent pages, the results in Table 3.3 are identical to the results obtained after navigational links elimination only. However, we had experiments where the number of near-duplicate parent pages was large enough to affect the results of Co-citation.

Input: a Web page given by its URL u.

1. Collect up to B parents of u.
2. Eliminate parents that point through a navigational link to the query URL u.
3. Detect the candidates for the near-duplicates contraction at the page level.
4. Merge the candidates' pages.
5. Eliminate siblings that are pointed to through a navigational link by their parents.
6. Compute the degree of co-citation for the remaining siblings.

Output: return as related pages the pages with the 10 highest degrees of co-citation.

Figure 3.5: Navigational links elimination and near-duplicates contraction.

We investigated the reasons for which the algorithm is still returning siblings that are only tangentially relevant to the query URL. We have discovered that, for many parents, the collection of links that we select as siblings of the query URL is not necessarily related to the query URL. This is the most important problem that we have identified and we discuss it in the next section.

3.4 "Unfocused" collection of links

Although it has been noticed that links that are close to each other on a page tend to be on a similar topic, a page may contain links related to several topics in several groups. By simply taking BF links immediately surrounding the query URL on a parent it is possible to collect links that span over more than one topic. To improve the results we would like to restrict the surrounding links to the group or topic that the query URL belongs to. Such groups have been called "pagelets" [11], and we adopt for our application an algorithm that is given in [2] and discussed in the "Related Work" chapter.

This algorithm considers HTML elements such as tables, paragraphs, headings, and lists to identify pagelets. Such an HTML element in the parse tree

Table 3.3: Original Co-citation - Results after navigational links elimination and near-duplicates contraction

URL	DESCRIPTION
www.ubc.ca	UBC homepage
www.psych.ubc.ca	UBC Psychology Dept.
www.linguistics.ubc.ca	UBC Linguistics Dept.
www.philosophy.ubc.ca	UBC Philosophy Dept.
www.ec-o.com	Internet - related Company
www.bsdi.com	Internet Services
www.cs.ubc.ca/nest/lci	UBC CS Laboratory for Computational Intelligence
www.ams.ubc.ca/clubs/cogsys	UBC Cognitive Systems Society
www.cs.ubc.ca/nest/imager/imager.html	UBC CS Imager Laboratory
life.anu.edu.au:80	Bioinformatics Group, Australian National University

of a page p is defined as a pagelet if

1. None of its children contains more than k hyperlinks, and

2. None of its ancestor elements is a pagelet.

In our application, we break parent pages into pagelets using this algorithm (we have used k equals 4), and then consider as a parent for Co-citation the pagelet that contains the query URL (if there are several pagelets that contain the query URL, we consider the first such pagelet). In this way, we may increase the chance that the siblings will be on a similar topic as the query URL, and hence more likely related.

Extracting pagelets out of the parent pages causes the modification of the near-duplicate contraction operation. Because our final set of parents will consist of pagelets, we determine the candidates for near-duplicates contraction at the *page* level; however, we will perform the merging operation at the *pagelet* level. When we encounter two parent pages satisfying formulas (3.1) and (3.2), we will merge their corresponding pagelets into a single pagelet, whose set of links is the disjoint union of the two pagelets.

We summarize the sequence of techniques that we perform on top of the original Co-citation algorithm in order to deal with the above-identified and discussed problems, i.e., (1) navigational links, (2) near-duplicate pages, and (3) pages with links related to multiple topics. We call the series of these operations "preprocessing". In the next chapters, we will often use the term Co-citation to refer to the preprocessed Co-citation algorithm. The preprocessed Co-citation algorithm represents the baseline against which we compare our focused versions of Co-citation. When we refer to Co-citation as proposed in [15] and presented in chapter 2, we will use the term *original* Co-citation. The pseudo-code for the preprocessed Co-citation algorithm is presented in Figure 3.6.

Input: a Web page given by its URL u.

1. Collect up to B parents of u.
2. Eliminate parents that point through a navigational link to the query URL u.
3. Extract a pagelet out of each of the remaining parents.
4. Detect the candidates for the near-duplicates contraction at the page level.
5. Merge the candidates' corresponding pagelets.
6. Eliminate siblings that are pointed to through a navigational link by their parents.
7. Compute the degree of co-citation for the remaining siblings.

Output: return as related pages the pages with the 10 highest degrees of co-citation.

Figure 3.6: Preprocessed Co-citation.

We have run the preprocessed Co-citation algorithm on our initial example. The results are presented in Table 3.4.

We notice some improvement in the results: we obtained two *homepages* of other computer science departments at positions six, and ten, and another university *homepage*, at position eight. Another department within the UBC

Table 3.4: Original Co-citation - Results after navigational links elimination and duplicates contraction and pagelet extraction

URL	DESCRIPTION
www.ubc.ca	UBC homepage
www.psych.ubc.ca	UBC Psychology Dept.
www.philosophy.ubc.ca	UBC Philosophy Dept.
www.tc.cornell.edu:80/ctc.html	Cornell Theory Center
www.bsdi.com	Internet Services
www.csc.uvic.ca	CS Dept., Univ. of Victoria
life.anu.edu.au:80	Bioinformatics Group, Australian National University
www.acns.nwu.edu	Northwestern Univ., USA
www.math.ubc.ca	UBC Mathematics Dept.
www.cs.toronto.edu	CS Dept., Univ. of Toronto

domain appears at position nine. One URL that did not occur in the previous results appears in Table 3.4, ranked forth, but it is marginally related to the query URL.

We consider that the results have not improved significantly overall.

3.5 The need for "focus"

Obviously, the extraction of pagelets based on the structure of the HTML page does not guarantee that the constructed pagelet is in fact a collection of topically related links. We claim that the notion of a pagelet is to weak to support Co-citation, because there are many collections of neighboring links that satisfy the pagelet definition, but are not on similar topics (e.g. unstructured bookmark lists). We call such pages and pagelets for which the majority of outgoing links point to pages on different topics "unfocused". When the number of "unfocused" parents for a query URL is a large fraction of the total number of parents, many siblings on different topics will have a large degree of co-citation with the query URL and the results will be distorted.

We argue that in order to produce better results we have to give more focused collections of links a higher influence on the final ranking than less focused collections. Therefore, we need to measure how well a collection of

links is focused on a topic. Three proposals to define "focus" will be presented in the next chapter.

Although we have seen just one example that supports our claim, our experiments in the last chapter show that the problem of unfocused parents is not a rare event and applying our focused version of the Co-citation algorithm will improve the results.

3.6 Preprocessing - Discussion

In this chapter we have reviewed a number of problems that may distort the results of the original Co-citation algorithm. To address these problems, we have applied a sequence of techniques, illustrated by a step-by-step example. In the case of *www.cs.ubc.ca*, after applying all the preprocessing techniques, we have obtained a slight improvement in the results of the original Co-citation.

We have analyzed the role that the preprocessing stage is playing on our set of experiments. For some experiments, the results of preprocessed Co-citation showed moderate improvement compared to the results of original Co-citation. However, we have encountered experiments for which the preprocessing stage slightly decreases the quality of the results obtained by the original Co-citation. We give an example of a query URL for which this phenomenon occurs. The query URL is *www.freshwasabi.com*. This URL is the Web page of the "Pacific Farms" company, that produces and sells online wasabi and wasabi products (wasabi is a condiment traditionally served with sushi and noodle dishes in Japan). We expect as related pages other Web sites that sell mainly Asian foods online.

The results obtained by Original Co-citation are presented in Table 3.5 and the results obtained by Preprocessed Co-citation are shown in Table 3.6. The last field indicates the relatedness of the result to the query URL (we use 1 for related, 0 for unrelated).

Original Co-citation and Preprocessed Co-citation share four results, out of which, three are on the topic of the query URL. In the case of the Preprocessed Co-citation only these three results are related, where Original Co-citation

Table 3.5: Example: Original Co-citation

URL	DESCRIPTION	Relatedness
www.wasabi.co.nz	Sells wasabi online	1
www.stickyrice.com	Sells sushi products online	1
www.bento.com/ tokyofood.html	Guide to eating in Tokyo	0
www.shoretodoor.com	Sells seafood online	1
www.importfood.com	Sells a variety of Asian foods online	1
www.orientalpantry.com	Sells oriental foods, exotic spices online	1
www.norpac.com	Sells fruits and vegetables online	0
www.fish2go.com	Sells seafood online	1
www.unclebens.com	Uncle Ben's rice	0
www.digitalsushi.net	Sushi related page	0

Table 3.6: Example: Preprocessed Co-citation

URL	DESCRIPTION	Relatedness
www.wasabi.co.nz	Sells wasabi online	1
www.uwajimaya.com	Asian foods market	0
www.shoretodoor.com	Sells seafood online	1
www.stickyrice.com	Sells sushi products online	1
www.eat.com	All kinds of recipes online	0
www.japantimes.co.jp	Newspaper: Japan Times	0
www.sake-world.com	Sake related page	0
www.bento.com/ tokyofood.html	Guide to eating in Tokyo	0
www.garden-gifts.com	Japanese garden gifts	0
www.japanesegifts.com	Japanese gifts	0

produces three more related results. This experiment shows that are cases when preprocessing may adversely affect the results of the original Co-citation. Our preprocessing stage is comprised essentially of a set of heuristics that in some cases have the expected effect and in other cases they have a different effect. The most disputable heuristic is the one that extracts pagelets out of parent pages. This heuristic produces the set of siblings that we consider in our setting. The relatedness to the query URL and the frequency of these siblings greatly influence the outcome of Co-citation. The authors of Web pages write their HTML pages in a variety of ways, so that heuristics that extract pagelets based on the HTML structure are prone to errors. On the other hand, selecting BF links surrounding the query URL may not also work well in all cases.

Given the fact that the effect of preprocessing stage can not be predicted accurately, we will embed our focus both within the original and the preprocessed versions of Co-citation. The next chapter defines the notion of "focus" and formalize it in several ways.

Chapter 4

Focused Co-citation

In this chapter, we formalize the notion of "focus" of a collection of links and exploit this notion in finding related pages.

Intuitively, the focus of a collection of links should capture the degree of agreement, in terms of a topic, between the corresponding pages in the collection. For example, a page containing a list of the Canadian computer science departments will be more focused than a bookmark-list on a homepage which points to pages on different topics including the UBC computer science department.

We formalize the notion of "focus" of a collection of links in three ways: a link-based, a content-based, and a hybrid approach. The link-based approach is based only on the linkage data that we already collected for the Co-citation algorithm, and therefore, is computationally inexpensive. The content-based approach takes into account the textual content of the siblings and of the query URL. The last method is a combination of the previous two, which attempts to balance the potential drawbacks of each of the link-based focus and the content-based focus.

This chapter is structured as follows. First, we discuss and define a ranking function, which will be used to rank the siblings of the query URL. Then, we formalize three types of focus: *LinkFocus*, *ContentFocus*, and *HybridFocus*. For each type of focus, we discuss potential problems and provide meaningful examples.

4.1 Ranking Function

Let us assume that we have a method for computing the "focusedness" of a collection of links. Let *Focus (A)* denote the computed focus for the collection of links A. In our case, we compute the focus of each of the parents existing in our data set, where a parent is actually a pagelet (which could be regarded as a collection of links).

In the Co-citation ranking scheme, the relatedness of a page s is computed according to the number of common parents with the query URL, as the proportion

$$Rank(s) = \frac{||CP||}{||AP||} \tag{4.1}$$

where CP is the set of common parents for the page s and the query URL, and AP is the set of all parents. For a set M, the notation $||M||$ denotes the number of elements of the set.

The purpose of computing a "focus" score for each of the parents is to recognize those parents that contain indeed siblings related to the query URL. We can compute a weighted version of the Co-citation ranking. The idea is to give focused parents a higher weight in the computation, so that in the end, the siblings descending from focused parents will have a higher rank than other siblings.

We propose a weighting scheme that will adjust the co-citation rank proportional to the "focusedness" of the common parents of the sibling s, i.e., the more focused parents a sibling has in common with the query URL, the higher the sibling's rank.

$$Rank(s) = \frac{\sum_{P \in CP} Focus(P)}{\sum_{P \in AP} Focus(P)} \tag{4.2}$$

The simple Co-citation ranking scheme is embedded in our ranking scheme: in the case of Co-citation, we could consider that all the parents have the same focus score, i.e., the focus is equal to 1. If we substitute *Focus (P)* with the value 1 in formula (4.2), we obtain exactly the Co-citation's ranking scheme:

$$Rank(s) = \frac{\sum_{P \in CP} 1}{\sum_{P \in AP} 1} = \frac{||CP||}{||AP||}$$

We also notice that in formulas (4.1) and (4.2) the denominator is a constant value: the number of all parents is a constant for a given data set, as well as the sum of the focus scores of all the parents. We divide in formulas (4.1) and (4.2) by a constant value as a normalization operation and because the formulas look more intuitive when presented in this way. However, the ranking of the siblings will not change even if we do not divide by the constant factor.

In our experiments, we used formula (4.2) to rank the results of the Co-citation algorithm and its variants. From this point on, when we talk about ranking scheme, we understand the ranking scheme given by formula (4.2).

There is another issue involved in the computation of rankings that might affect the quality of the results. It is possible to have an unrelated sibling that has a large number of parents in common with the starting URL and all of these parents have a low focus. However, because the number of common parents is very high, this sibling might get a high rank in the computation – maybe not as high as in the co-citation ranking, but still high.

A potential sibling of this type is a "popular" URL. By "popular", we understand those URLs that have a very large in-degree and that are unrelated to most other pages. Popular siblings (URLs) will get high ranks, because they are pointed to by a large fraction of parents out of the total number of parents and even if these parents have low focus, when we sum up these focus scores, we end up with a sufficiently high value, which translates into high ranks.

In order to avoid the problem of popular siblings, we have constructed manually a list of popular URLs such as *www.yahoo.com* , *www.adobe.com* , etc. When the query URL is not a popular URL itself, we eliminate these "popular" URLs from our set of siblings, because they are unrelated to the query URL. If the query URL is a popular URL itself, then we keep the popular siblings, because they might be in fact related.

In order to apply our new ranking function, we have to define a notion of focus of a collection of links. We formalize and discuss three new notions of focus: *LinkFocus* (section 4.2), *ContentFocus* (section 4.3) and *HybridFocus* (section 4.4).

4.2 LinkFocus

Let us assume that we have a web page or a pagelet, represented as a collection of links. Each of these links points to a web page and we want to assess how topical-cohesive this set of pages is. We will use a similar intuition as for co-citation in defining the first type of focus, that we call *LinkBased* focus.

Let us consider the example presented in Figure 4.1. Assume that we have a collection A composed of three links: L_1 that points to page p_1, L_2 that points to page p_2 and L_3 that points to page p_3. We want to estimate to what degree the pages p_1, p_2 and p_3 agree on a topic.

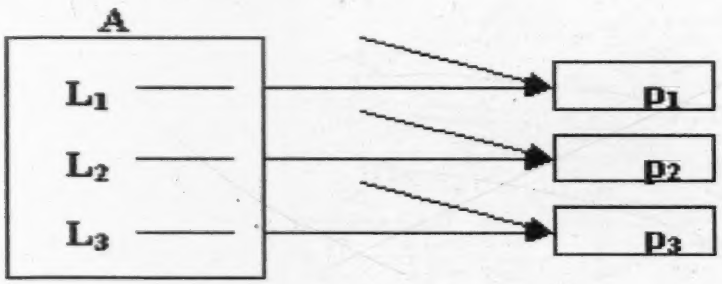


Figure 4.1: LinkBased Focus

Each of the pages p_1, p_2, and p_3 might have other parents except A. Let P be the disjoint union of the parents of p_1, p_2,and p_3. The elements of P are not illustrated in the Figure 4.1, but arrows indicate their presence.

According to co-citation (which measures a certain notion of similarity of *pairs* of pages) the more parents two pages have in common, the more likely they are on the same topic. For example, the higher the number of common parents for p_1 and p_2, the more similar p_1 and p_2 according to co-citation. For our purposes, we want to extend this notion of similarity to a whole *set* of pages that may contain more than two pages, i.e., we want to assess the similarity of the *set* comprised of p_1, p_2, and p_3 as a whole, not just the similarity of the pairs (p_1, p_2), (p_2, p_3), and (p_3, p_1) separately.

The intuition is that the more parents exist in P that agree on more of A's children (in this case p_1, p_2 and p_3), the more focused the collection of links A

will be. The collection of links A will have maximum focus in the case when all the parents in P agree on all of A's children (i.e., all the parents in P share the links to p_1, p_2 and p_3). The collection of links A will have minimum focus in the case when none of the other parents in P share more than one link with A, i.e., any other parent in P will share either the link to p_1, or the link to p_2, or to p_3, but it will never share a combination of two or three of them.

According to our intuition, any parent B from P that shares i links with A should contribute to the focus of A proportional to the number of shared links. For example, if B and C are parents from P so that B shares with A the links to p_1, p_2, p_3, and C shares with A the link to p_1, then B should contribute more than C to the focus of A. The contribution of B to the focus of A should be proportional to the number of shared links, i.e., the number $||$ A $\cap$ B $||$, which in our example is 3. Similarly, the contribution of C to the focus of A should be proportional to $||$ A $\cap$ C $|| = 1$.

However, we also have to take into consideration the number of links that A and B have individually. The reason is that pages with a large number of links have a higher chance of sharing one or more links with any other page. If, for instance, B has a large number of links and shares with A one or two links, then the contribution of B to the focus of A should be lower than if B had a small number of links. Similarly for the case when A has a large number of links. The contribution of a parent B to the focus of A should be proportional to the number $|$ A $\cup$ B $|$, i.e., the number of links in the disjoint union of A and B.

We formalize this intuition in the following definition of the focus of a collection of links A:

$$LinkFocus(A) = \sum_{B, B \neq A} \frac{||A \cap B||}{||A \cup B||} \tag{4.3}$$

where $||.||$ denotes the number of elements of a set and B is any other parent from P.

In the case of the Co-citation algorithm, we have collected the parent pages of the query URL (up to a fixed number). By the way we construct our data set, each parent points to the query URL, so that any two parents in the Co-

citation scheme will share at least one link, i.e., the link to the query URL page. We compute 4.3 for each of the parents A existing within our set of parents.

Formula (4.3) could be regarded as a "neighborhood" function, measuring the "degree of clustering" between a parent pagelet A and all other pagelets B that are also parents of the query URL. The fraction $\frac{||A \cap B||}{||A \cup B||}$ is always less than or equal to 1, for any A, B parents. However, *LinkFocus(A)* could be greater than 1. We normalize *LinkFocus(A)* to be a value between 0 and 1, so that the focus of a collection of links A is given by the following formula:

$$LinkFocus(A) = \frac{\sum_{B, B \neq A} \frac{||A \cap B||}{||A \cup B||}}{||AP||} \tag{4.4}$$

where P is the set of all parents of the query URL, $||.||$ denotes the number of elements of a set and B is any other parent from P.

Computing the link-based focus, as defined in (4.4), for our data set is computationally very cheap, because we already have stored the data set and we have computed all the necessary values in the standard computation of Co-citation.

We also notice the relationship between the near-duplicates contraction performed in the preprocessing step and the computation of *LinkFocus*. As we explained in the previous chapter, the near-duplicates detection is done at the page level. For every parent page, we extract a pagelet, so that in our setting, the parents are actually pagelets. For the pages that are deemed candidates for near-duplicates contraction, the corresponding pagelets are merged. In formula (4.4), A and B are the extracted pagelets. It is possible to have high values for the fraction $\frac{||A \cap B||}{||A \cup B||}$, as high as 1 (for the case when the extracted pagelets are identical). This fact does not imply that the focus of A is artificially high because of mirroring across multiple sites. This fact rather implies that the focus of A is high because many other parents agree on many of A's children, i.e., A is a *focused* collection of links.

We applied the link-based focus to our running example, and the results are shown in Table 4.1. The *LinkFocus* is computed on the preprocessed data,

according to formula (4.4). The ranking of the siblings is obtained by applying the ranking scheme described in section 4.1.

Even though link-based Co-citation is over all the experiments slightly better than Co-citation, as shown in chapter 5, unfortunately, in this particular example, the results for the link-based focus combined with Co-citation are almost identical with the results of Co-citation shown in Table 3.4.

Table 4.1: LinkFocus

URL	DESCRIPTION
www.ubc.ca	UBC homepage
www.psych.ubc.ca	UBC Psychology Dept.
www.philosophy.ubc.ca	UBC Philosophy Dept.
www.tc.cornell.edu:80/ctc.html	Cornell Theory Center
www.bsdi.com	Internet Services
www.math.ubc.ca	UBC Mathematics Dept.
www.chem.ubc.ca	UBC Chemistry Dept.
www.acns.nwu.edu	Northwestern Univ., USA
life.anu.edu.au:80	Bioinformatics Group, Australian National University
www.cs.toronto.edu	CS Dept., Univ. of Toronto

This example shows us, however, that linkage information may not be enough to compute effectively the focus of a collection of links in all cases.

There are two issues that need to be addressed at this point. The first one is that the results of the *LinkFocus* are similar to the results obtained by Co-citation, as we can see from Tables 3.4 and 4.1. This is not surprising, given the fact that both methods are based solely on linkage information and that the very idea of the *LinkFocus* has at its roots the co-citation technique.

The second issue that we discuss is why *LinkFocus* does not seem to perform very well in the above example. We have analyzed our data set and we have discovered what we call "*parents in conspiracy*", i.e., a set of parents that contain the same collection of links and these links are not necessarily related to the query URL. If the number of parents in the "conspiracy" is sufficiently large, then Co-citation will rank the siblings in the repetitive collection of links high, because these siblings have a sufficiently large number of common

parents with the query URL. At the same time, *LinkFocus* will rank this set of siblings also high, because there are a large number of parents that agree on a sufficiently large number of children. How strong this "conspiracy" phenomenon is manifested in our data set depends on several factors, such as the fraction of parents in the conspiracy out of the total number of parents, the size of the repetitive collection of links and the actual number of links of the parents in our data set.

The second issue from above assumes that the collection of links that is shared by the parents in the conspiracy is not comprised of URLs related to the query URL. If this repetitive collection contains URLs indeed related to the query URL, then this "conspiracy" is what we are looking for, so that Co-citation and, subsequently, *LinkFocus* will produce good results. The challenge is to distinguish between the cases when this conspiracy is what we need and the cases when it is not. This distinction cannot be achieved based on the linkage information only, because the linkage pattern is the same in both cases.

We propose another notion of focus, *ContentFocus*, based on textual information, that attempts to solve the problems associated with methods based solely on the linkage structure of the Web.

4.3 Content Focus

As we stated before, the focus of a collection of links captures the degree of agreement on a topic amongst the pages pointed to by the links within the collection. This is a general formulation of the "focusedness" of a collection of links that does not make any assumption about the topic with respect to which we compute the focus. However, in the case of the Co-citation algorithm, we are interested in finding pages that are on the query URL's topic. It follows that for the Co-citation algorithm we want to compute the focus of each of the parents within our data set with respect to the topic of the query URL. This fact is embedded already in the LinkFocus computation (see (4.4)), because, by construction, each of the parents within our data set contains a link to the query URL.

Let us consider a parent A of the query URL within our data set (see 4.2). We aim to collect textual content for each of the links within A as indicative as possible for the topic of the page pointed to by the link. The goal is to formalize a notion of focus based on the collected textual information that measures the degree of agreement in terms of the query URL's topic of the pages pointed to by the links existing in A.

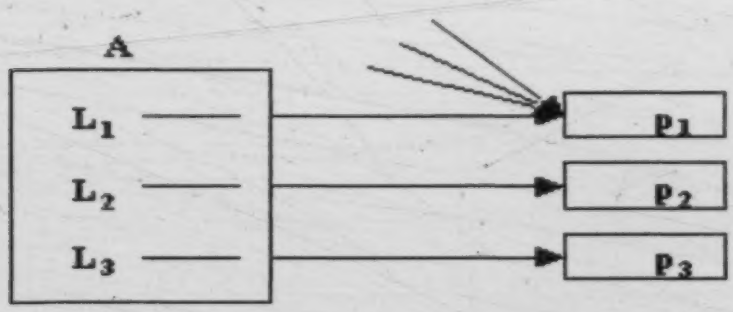


Figure 4.2: ContentBased Focus

For each link in A, we collect textual content in the following manner: from the page the link points to, we extract the title, and we concatenate it with the union of the anchor texts of all the incoming links of the page. The incoming links of page p_1 are depicted by arrows pointing to p_1. The incoming links of p_2 and p_3 are omitted for clarity. We restrict the set of incoming links to the set of parents of the query URL collected by the Co-citation algorithm. In order to collect the textual content of link L_1, for instance, we parse the page pointed to by L_1, i.e., p_1, and we extract from it the title. Then, for all the parents in our data set that point to p_1, we collect the anchor-text of the link to p_1. The union of these anchor-texts and the title of p_1 comprise the textual information collected for L_1.

We consider anchor text when collecting the textual information for a link because anchor text has been successfully used as an indication of the topic of the page pointed to by the link in other works ([21], [12], [20]). Some existing work uses instead of anchor text a fixed window around the anchor text, including the anchor text [21]. In our work, we have considered only the

anchor text of a link and it proved to perform well for our purposes.

With respect to our data set, we consider the set of all the siblings of the query URL and the query URL itself. Let this set be S. For each URL s in S, we collect textual content by collecting the anchor text of all the incoming links of s and the title of the page s points to. The title of s may be a good indication of the topic of s. However, we have two other alternatives, in which we collect increasingly more textual content from s, i.e., we consider "meta" description and headings. To summarize, for all s ∈ S, we collect textual content in three different ways:

- *Approach 1.* Union of the anchor text of all the incoming links of s and, from the page s points to, the title.
- *Approach 2.* Union of the anchor text of all the incoming links of s and, from the page s points to, the title and the "meta" description.
- *Approach 3.* Union of the anchor text of all the incoming links of s and, from the page s points to, the title and the "meta" description and the headings.

We want to assess the topic of s, for all s ∈ S, based on the collected text. The experiments in chapter 5 will show which of these three approaches is the most effective one. However, independent of the approach used for collecting the text, we will manipulate the resulted textual information in the same manner.

We eliminate stop words and we perform a light stemming (the plural endings are eliminated, as well as the endings –ed, and –ing) on the text collected for all s ∈ S. We added to the list of stop words a few words that are very frequent on the Web and do not carry any meaning with respect to any topic. For example, given the fact that we collect anchor texts, we have added "click", "here" to the list of stop words, because these two words could appear frequently within the anchor text of a link and do not convey anything regarding the topic of the page pointed to by the link. Other examples of "Web-specific" stop words are"link", "online", "website" etc.

We will use the vector space model for representing the textual content of the siblings and of the query URL. The vector space model assumes a collection of documents and a universe of words so that each document is represented as a vector of length the dimension of the universe of words. These vectors consist of *weights*, i.e., values that represent the "importance" of the words within the universe of words with respect to the textual content represented by the vector.

In our case, a "document" is the text collected for any s in S. Given the fact that we want to measure the "focusedness" of a parent with respect to the query URL's topic, the universe of words will be formed by the words (or *keywords*) extracted when parsing the text collected for the query URL. Each $s \in S$ will be represented as a vector in this universe of keywords.

One issue that needs to be addressed is what are the weights in the resulting vectors. Each location within a vector corresponds to a distinct keyword. A simple idea would be to define the weight as the frequency of the keyword within the document represent by the vector. The intuition is that if a keyword appears many times in a document, then probably the document is on the topic suggested by that keyword. We have already eliminated stop words, such as "and", "on", etc., because these words are frequent in any document and do not convey anything about the topic of the document. This weight is called "term frequency", and noted with tf.

Research in the classic Information Retrieval field has shown that the tf weighting scheme is not always the best one. Imagine that we have a collection of documents from a conference on Web related issues. The keyword "web" is probably frequent within each document and within the entire collection; however this keyword is not very informative if we want to distinguish between documents on different sub-areas or topics, since we already know that all the documents are about the Web. Another weighting scheme, called $tf - idf$ scheme, has been proposed, which takes into account how frequent a keyword is, not only within a given document, but also within the entire collection. There are several variations of the $tf - idf$ scheme. The following formula is often used to compute the weight of each location within the vector

representing a document in the $tf - idf$ scheme:

$$weight = tf * \log_{10}(\frac{N}{df}) \tag{4.5}$$

where:

- tf is the term frequency.
- N is the number of documents existing in the collection, i.e., in our case, is the number of elements of S.
- df or the " *document frequency*" is the number of times a given term appears across all the documents in the collection. The number $\frac{N}{df}$ is referred usually as the "*inverse document frequency*" and noted by *idf*". From here the name of the scheme.

The $tf - idf$ scheme decreases the weight of keywords with high document frequency and amplifies the weight of keywords with low document frequency. In the formula (4.5), if a keyword occurs in many documents within the collection (i.e., its document frequency, df, is close to N), then the idf value is close to 1, so it follows that the weight given to the keyword is close to 0 ($log_{10}1=0$).

Increasing the weight of keywords with low document frequency might be in fact useful for ad-hoc queries, where a rare keyword in the query should receive the most importance. However, if we want to evaluate document similarities, rare terms may not be useful at all, because they might be typos (especially on the Web), rare names or non-topical terms that might hinder the similarity measure. Other variations of the $tf - idf$ scheme that attenuate both high and low document-frequency terms may be used instead of weighting schemes such as the one in (4.5) for judging document similarities ([21]).

In our case, each sibling is represented as a vector in the universe of keywords extracted for the query URL. As we will explain below, for our Content-Focus, we want to distinguish the "closest" siblings to the topic of the query URL. Because this "closeness" will be computed based on the vector representations of the sibling and the query URL, it is important to decide how

to compute the weights that will populate the vectors and what similarity function to use.

The way we collect the text for the query URL is aimed to obtain keywords that are as informative as possible for the topic of the query URL. However, in practice, not all the extracted keywords are meaningful for the topic of the query URL. Some keywords are general enough so that they become related to any topic (e.g., "item"). Other keywords are typos or words from other language than English (e.g., "departamento").

We tried to judge if the $tf - idf$ scheme would be suitable for our purposes, i.e., if the meaningful keywords are likely to have high document-frequency or not. We have examined the keywords with the highest document frequencies for a number of experiments and we have noticed that these keywords are meaningful for the topic of the query URL. Some examples are presented in Table 4.2 (the keywords are ordered descending after document-frequencies).

Table 4.2: Frequency of keywords

URL	TOPIC	High-Frequency Keywords
www.cs.ubc.ca	Computer Science Dept.	university, science, computer, department
www.oktoberfest.ca	Festival held in Kitchener-Waterloo	canada, waterloo, festival, kitchener
www.synquest.com	Supply Chain Management	chain, supply, performance, viewlocity
www.antiqueradio.com	Buyers and Sellers of Old Radios and Related Items	radio antique vintage collector
www.planettribes.com/tribes2	Game on "tribes", part of the GameSpy network	tribe planet gamespy network

Table 4.2 suggests that meaningful keywords may have high document-frequency within our data set. Besides such empirical observations, the way we have collected and preprocessed the data (i.e., the extraction of pagelets is aimed to produce siblings that are as close as possible to the query URL's topic) drives us to the conclusion that "important" keywords are likely to have high document-frequency, so the $tf - idf$ scheme is not suitable for our purposes.

Another reason for which we rejected the use of the $tf-idf$ scheme and even the use of the tf scheme descends from the way we compute our ContentFocus. Intuitively, a "focused" parent will be a parent for which the majority of its children (or siblings with respect to the query URL) are "close" to the query URL in terms of the vector representations. We give an example to clarify our decision. Let the frequency vector representing the query URL be 3,2,1, i.e., the first term appears 3 times, the second term, 2 times, and the last term, 1 time, within the text collected for the query URL. Suppose the sibling s_1 is represented by 1,1,5, and the s_2 is represented by 1,1,7. In terms of cosine similarity measure, the sibling s_1 is much closer than the sibling s_2 to the query URL. Let us assume that the keywords with frequencies 3 and 2 are the informative ones for the topic of the query URL and the last keyword is the bogus one. With regard to the query URL's topic, the siblings s_1 and s_2 are equivalent. However, it is possible that parents that point to s_1 will receive a higher focus score than parents pointing to s_2 and yet, this higher focus value does not reflect the true "focusedness" of the parent with respect to the query URL's topic. The final effect is that the results will not be on the query URL's topic, but rather they will "diffuse" to other, possible related, topics.

The assumption that the keywords with high frequency are the most representative for the topic of the query URL is likely to be true in many cases. The reason is that we considered anchor text when we collected the text for the query URL: if the majority of many independent authors refer to the query URL by using the keyword, for instance, "university", than probably the topic of the query URL is within the academic area. Determining automatically which keywords are indeed meaningful for the query URL's topic, in the general case, is difficult, due to the particularities of the Web. One idea would be to select as keywords only the keywords having the frequency above a certain threshold. In some cases, only the highest frequency keywords could guarantee that no diffusion phenomenon occurs; in other cases, we need to go as low as frequency 1, because, for instance, the majority of the authors used as anchor text the words "click here". We will discuss more on the issue of selecting the informative keywords in the next chapter.

In order to circumvent the above explained problem, we use the binary weighting scheme: if a keyword appears within a document, its weight is 1, otherwise its weight is 0. In this way, the influence of bogus keywords is equalized for all the siblings. For our purposes, the presence of a keyword within the text collected for a sibling is enough to consider the sibling on the topic given by the keyword.

We use the Jaccard matching coefficient as a similarity function. Given two vectors of the same length, v_1 and v_2, consisting of only 0 and 1 values, we note with:

- p - the number of *positive matches*, i.e., the number of entries on which both vectors have the value 1
- s - the number of entries on which the first vector has the value 1 and the second vector has the value 0
- r - the number of entries on which the first vector has the value 0 and the second vector has the value 1
- n - the number of *negative matches*, i.e., the number of entries on which both vectors have the value 0

The Jaccard matching coefficient is given by the following formula:

$$Jaccard(v_1, v_2) = \frac{p}{p + r + s} \tag{4.6}$$

In our case, we only want to know if a keyword occurs within the text collected for a sibling, so that the Jaccard matching coefficient is indeed suitable, because it considers the number of negative matches unimportant and therefore, ignored.

We formalize ContentFocus in the following way. If A is a parent page, comprised of the links $L_1, L_2, \ldots, L_{M-1}$, and L_0 (the link to the query URL), let $v_1, v_2, \ldots, v_{M-1}$ and v_0 be the vectors corresponding to these links. The intuition for ContentFocus is that the more similar $v_1, v_2, \ldots, v_{M-1}$ are to v_0, the higher the focus of A. We formalize this idea in the following formula for

the *ContentFocus* of a parent page A:

$$ContentFocus(A) = \frac{\sum_{i=1}^{M-1} Jaccard(v_i, v_0)}{M-1} \tag{4.7}$$

where $M-1$ is the total number of links of A, except the link to the query URL, and $Jaccard(v_i, v_j)$ is computed according to formula (4.6), for all i from 1 to $(M-1)$.

Formula (4.7) tries to capture the average similarity of a set of points with respect to a given point.

We note that:

$$0 \leq Jaccard(v_i, v_0) \leq 1, \forall i = 1, \ldots, (M\text{-}1) \Rightarrow$$
$$0 \leq \sum_{i=1}^{M-1} Jaccard(v_i, v_0) \leq M-1 \Rightarrow$$
$$0 \leq ContentFocus(A) \leq 1$$

Regardless of the approach used for collecting the textual information, the computation of the ContentFocus is the same. The length of the vectors and the weights change depending on how much text we consider. We vary the amount of the text in our computations and experiment with:

- *ContentFocus1*: the *ContentFocus* when the textual content is gathered according to Approach 1.
- *ContentFocus2*: the *ContentFocus* when the textual content is gathered according to Approach 2.
- *ContentFocus3*: the *ContentFocus* when the textual content is gathered according to Approach 3.

We have applied the *ContentFocus* to our running example. The results in Table 4.3 are obtained by computing the *ContentFocus1* on preprocessed data.

We can notice a significant improvement in the results. We have obtained five *homepages* of other computing science departments from Canada, the *homepage* of the UBC together with four *homepages* of other departments

Table 4.3: ContentFocus

URL	DESCRIPTION
www.ubc.ca	UBC homepage
www.psych.ubc.ca	UBC Psychology Dept.
www.philosophy.ubc.ca	UBC Philosophy Dept.
www.csc.uvic.ca	CS Dept., Univ. of Victoria
www.cs.toronto.edu	CS Dept., Univ. of Toronto
www.math.ubc.ca	UBC Mathematics Dept.
www.csd.uwo.ca	CS Dept., Univ. of Western Ontario
www.chem.ubc.ca	UBC Chemistry Dept.
www.cs.umanitoba.ca	CS Dept., Univ. of Manitoba
www.cs.uregina.ca	CS Dept., Univ. of Regina

from UBC. The first three results were obtained steadily by Co-citation and LinkFocus, so that we infer that there are a large fraction of parents that cite them. ContentFocus returns other departments from UBC, because of keywords specific to the UBC domain, such as "british", "columbia", or "ubc", that are frequent within our collection of documents and drive the results of the ContentFocus towards other departments from the UBC domain.

Regarding ContentFocus2 and ContentFocus3, the results for the running example are the same as the one presented in Table 4.3. We will discuss the effect of collecting increasingly more text in the "Experimental Evaluation" chapter.

4.4 HybridFocus

As suggested by the example presented in Table 4.3 and as we show in our experimental evaluation, the ContentFocus performs better than preprocessed Co-citation or LinkFocus. However, if the textual information that we collect is scarce or it can not be collected at all, then ContentFocus will face problems.

The HybridFocus is intended to balance the potential drawbacks of LinkFocus by using ContentFocus, and the vice-versa, to compensate the potential drawbacks of ContentFocus by using LinkFocus. This intuition could be formalized into the following general formula for the HybridFocus of a parent

A:

$$HybridFocus(A) = f * ContentFocus(A) + (1 - f) * LinkFocus(A) \quad (4.8)$$

Function f takes values between 0 and 1. We can notice that:

$$f = 1 \Leftrightarrow HybridFocus(A) = ContentFocus(A)$$
$$f = 0 \Leftrightarrow HybridFocus(A) = LinkFocus(A)$$

Any other value of f between 0 and 1 produces a combination of LinkFocus and ContentFocus.

The construction of the function f exploits the idea that the more content the A's children have, the more ContentFocus should count ($f \sim 1$), and the less content A's children have, the more LinkFocus should count ($f \sim 0$). This idea is based on the fact that ContentFocus performs better than LinkFocus, when we are able to extract "enough" textual information.

Let us assume A is a parent that contains a set of links. Each link is represented as a vector in the vector space model, where the universe of words is given by the keywords extracted for the query URL. Let k_1, k_2,...,k_n be the keywords extracted for the query URL, so that the dimension of the vector space model is n. The simplest situation is when no keyword has been extracted for the query URL. In this case, computing the ContentFocus for A is not possible, so that the only way of computing the focus of A is by using LinkFocus. It follows that:

$$n = 0 \Rightarrow f = 0 \Leftrightarrow HybridFocus(A) = LinkFocus(A)$$

If $n \neq 0$, then the larger n is, the closer to 1 f should be. After the point where n equals a certain threshold, how large is n does not matter anymore. For instance, for n equals 100 we will use ContentFocus only, as well as for n equals 200 or more. The following graph illustrates how f should behave, according to the number of extracted keywords, n.

We formalize the intuition that we have on f in the following formula:

$$f = min(log_T(n + 1), 1) \quad (4.9)$$

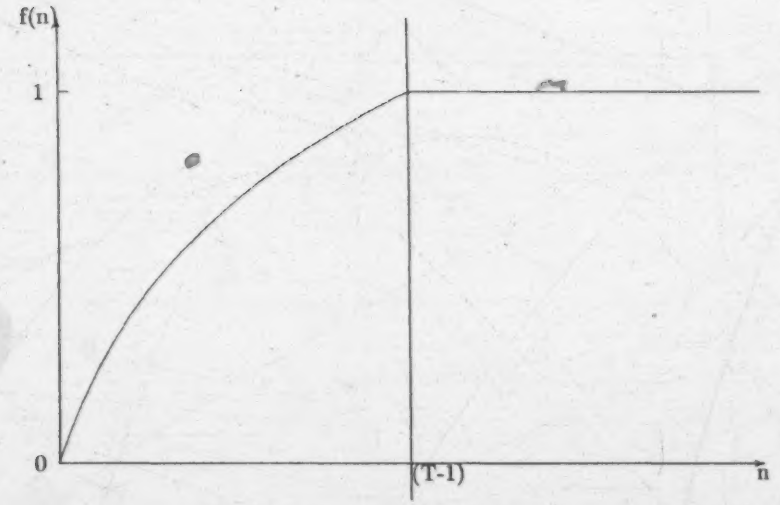


Figure 4.3: HybridFocus: joined Link and Content Focus

T ($T \geq 2$) is a constant value representing the threshold starting from which we will consider only the ContentFocus in (4.8). We show that f respects our intuition:

$(n = 0 \Leftrightarrow No\_keywords\_have\_been\_selected)$
$\Rightarrow log_T 1 = 0$
$\Rightarrow f = 0$
$\Leftrightarrow HybridFocus(A) = LinkFocus(A)$

$n \geq (T-1)$
$\Rightarrow log_T(n+1) \geq 1$
$\Rightarrow f = 1$
$\Leftrightarrow HybridFocus(A) = ContentFocus(A)$

$0 < n < (T-1)$
$\Rightarrow 0 < log_T(n+1) < 1$
$\Rightarrow f = log_T(n+1)$
$\Leftrightarrow HybridFocus(A) = f * ContentFocus(A) + (1-f) * LinkFocus(A)$

In (4.9), (T-1) is actually the threshold, i.e., the minimum number of keywords that we had selected for the query URL, so that the HybridFocus of a parent is in fact its ContentFocus. In our experiments we have used

$$(T-1) = 5 \Leftrightarrow T = 6$$

, i.e., if at least 5 keywords have been extracted, then HybridFocus is equivalent to ContentFocus, because we noticed empirically that 5 keywords are enough for the ContentFocus to perform well. However, this value is just a threshold that could be varied.

With regard to our running example, 20 keywords have been extracted for the query URL, so that the results obtained by HybridFocus are identical to the results obtained by ContentFocus (see Table 4.3).

4.5 Summary

Figure 4.4 illustrates the pseudo-code of Focused Co-citation, which is the same for link-based, content-based and hybrid focus and it is not depended on the application of the preprocessing stage.

Input: a Web page given by its URL u.

1. Build a *vicinity graph* for u.
2. For each parent of u compute a *focus* score.
3. Rank each sibling of u according to formula (4.2).

Output: return as related pages the pages with the 10 highest rank scores.

Figure 4.4: Focused Co-citation

In the next chapter, we will evaluate our methods, both on preprocessed and on un-preprocessed data and we show that these focused version of Co-citation constantly outperform the unfocused version with respect to the precision of the retrieved results.

Chapter 5

Experimental Evaluation

In this chapter we evaluate our methods by comparing the performance of the following six algorithms. The first algorithm is Co-citation, which acts as a baseline against which we will compare its focused versions. Because we want to show the performance of our methods regardless of the use of the preprocessing stage, we will present two sets of results. In the first set of results, the baseline algorithm is Preprocessed Co-citation (Figure 3.6). The Original Co-citation, as described in the "Related Work", chapter will be the baseline for the second set of experiments. No matter the flavor of Co-citation that we compare against, the next five algorithms illustrate the effect of our new notion, the "focus" of a collection of links, on the results of the Co-citation algorithm (Figure 4.4). The first algorithm, *LinkFocus*, shows the performance of the link-based focus, as formalized in section 4.2. The next three algorithms illustrate the content-based focus. When collecting textual content for each of the siblings and for the query URL, we have three alternatives that take increasingly more content into account (section 4.3), so we have three variations of the content-based focus: *ContentFocus1* (anchor text and title), *ContentFocus2* (anchor text, title and meta description) and *ContentFocus3* (anchor text, title, meta description and headings). We want to measure what is the improvement by taking increasingly more content into account. The last algorithm that we evaluate is the *HybridFocus*, as formalized in section 4.4. In our experiments we have used ContentFocus1 in the computation of the HybridFocus (see formula 4.8). Table 5.1 summarizes the algorithms that

we are evaluating.

Table 5.1: Evaluated Algorithms

ALGORITHM	DESCRIPTION
Co-citation	Original or Preprocessed
LinkFocus	Link-based Focus
ContentFocus1	Content-based Focus (anchor text, title)
ContentFocus2	Content-based Focus (anchor text, title, meta description)
ContentFocus3	Content-based Focus (anchor text, title, meta description, headings)
HybridFocus	Hybrid Focus

5.1 Implementation

For our methods, we need to obtain the in-coming links of the query URL. Several search engines provide this facility. We have examined several alternatives and we have decided to use Yahoo! [42] to obtain the in-coming links of a query URL. Google [20] might have been an option, but unfortunately, it does not allow automatic querying, unless authorized by a certain agreement. We have noticed that Altavista [1] and other search engines perform some preprocessing of the results of a query before presenting them to the user. Yahoo! served well for our purposes, since it allows automatic querying, does not alter the results of the *link* query and its results are presented in an easy-parsable format.

We have implemented a multi-threaded Java application that sends a *link* query to Yahoo!, parses the answers in order to obtain the URLs of the parent pages, and downloads the parent pages locally. For each parent page, the Java application downloads the content of its children. In this way, after running the Java application for a query URL, we have stored locally, in convenient data structures, the *vicinity* graph around the query URL that we need for our algorithms. The algorithms and other preprocessing steps are implemented in the *C* programming language (only the extraction of pagelets out of every

parent page is implemented in Java, due to the better facilities that Java has for parsing HTML files).

Downloading the necessary information for a query URL in Java may be time-consuming, depending on how dense the graph structure is around the query URL. However, once we have this information locally, the running time of the algorithms is low. We are interested in evaluating the effectiveness of our methods, rather than execution times.

5.2 Evaluation Strategy

The notion of "relatedness" is subjective and difficult to measure. "Relatedness" or "similarity" are usually measured by user studies. The users rate the results according to a given scale (often, a binary scale is used: 1 for related/similar, 0 otherwise) and then, based on the resulted scores, various metrics are computed.

When the goal is to evaluate a large number of experiments under different settings, user studies may not be suitable, given the fact that they are expensive both in time and resources. Instead, an automatic evaluation approach is preferred. Such an automatic evaluation assumes that there exists a "ground truth" for relatedness/similarity that could be used for evaluation. Web directories, such as the Open Directory (ODP) [30], have the potential of acting as a "ground truth" form for evaluating relatedness/similarity. Intuitively, the most related documents to a source document are the ones classified under the source's category, followed by documents classified under a sibling category, and so on. For instance, if the source document is classified under */Top/Arts/Movies/Awards*, the most related URLs are those under the same node of the directory tree, followed by documents in */Top/Arts/Movies/Film Festivals*.

Commercial directory sites usually have a small team of editors that evaluate the submissions and assign them to the right categories. However, given the rapid growth of the WWW, the quality and comprehensiveness of directories build in this manner has decreased. Open Directory Project (ODP)

proposes another approach where virtually anybody could volunteer as an editor, i.e., a person that maintains a certain category of interest (however, one could not add freely categories to ODP, unless approved by the ODP staff). Open Directory is the largest distributed database of Web content classified by humans and it constitutes the core of other directory services, such as Google [20]. ODP is an open source project, its data being available for download online.

We have used Open Directory both for the evaluation of the results and for the generation of the query URLs. To measure the effectiveness of our algorithms, we want to assess the percentage of "relevant" URLs returned by each algorithm. By "relevant", we understand

- URLs that are on the same topic as the query URL
- URLs that are good quality pages

Based on the principles Open Directory has been developed, it is very likely that the pages categorized inside are good quality pages, so condition (2) holds. Pages categorized under the same node are on the same topic, so that if a URL result is under the same node as the query URL, we will consider it "related" and we will score it with 1.

However, considering as related only the pages that are classified under the same node as the query URL is not suitable in practice due to the shortcomings of the Open Directory itself. First, the Open Directory is incomplete, i.e, pages that are relevant for a given category do not appear at all in the Open Directory. For instance, the query URL *www.cs.ubc.ca* is categorized under *Top/Computers/Computer_Science/Academic_Departments/North_America/Canada/*. URls that are related to the query URL, such as *www.csc.uvic.ca*, or *www.csd.uwo.ca*, are listed neither under this category, nor in any other node of the directory. Second, although in most of the cases pages are categorized based on topical considerations, there also pages that are grouped together due to geographical, rather than topical reasons (the branch *Top/Regional*). Last, it is possible to have a page classified under several categories. For example, *www.cs.ubc.ca* is categorized also under *Top/Reference/Education/*

Colleges_and_Universities/North_America/Canada/British_Columbia/University_of_British_Columbia/Departments_and_Programs/Science,_Faculty_of/, category which lists several other department from UBC.

Given these problems, we have used a "collapsed" version of the Open Directory tree for our evaluation, i.e., we have collapsed the directory below a fixed depth of three. Figure 5.1 illustrates the collapsed version of ODP.

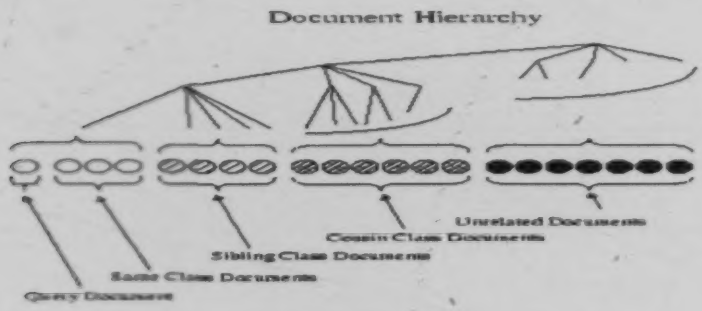


Figure 5.1: Collapsed Open Directory

The same approach has been used in [21], where the Open Directory was used as a ground truth to evaluate diverse similarity search algorithms on the Web. By collapsing the directory, we increase the chance of finding the results of our algorithms in ODP. On the *collapsed* directory, we consider as *related* pages, the pages that are in the same class as the query URL, and we score them 1; all the other pages are unrelated and we score them 0.

For our purposes, a query URL needs to be categorized somewhere in the Open Directory; otherwise we would not be able to evaluate if the results of our algorithms are related or not to it using ODP. The potential query URLs are chosen randomly from the Open Directory. A potential query URL is a valid query URL if it has at least 50 parents, because we want to make sure that there is enough linkage information around the query URL to obtain meaningful results. We have generated in this manner 100 URLs that we tested in our experiments.

5.3 Experimental Results

To estimate the performance of our algorithms, we use two measures that have also been used to evaluate the original Co-citation algorithm in [15]. The first measure is the *precision at R* for a given algorithm, which is defined as the total number of answers receiving a score of '1' within the first R answers, divided by R times the number of query URLs. The second one is the *average precision* for a given algorithm, which is defined as the sum of all the average precisions for all the query URLs, divided by the total number of query URLs. The average precision of a given algorithm with respect to a given URL u is the sum of the precisions at each rank where the answer of the algorithm for u received the score 1, divided by the total number of the answers of the algorithm for u receiving a '1' score.

We have computed *precision at R* and *average precision* for our six algorithms on both preprocessed and un-preprocessed data.

Figure 5.2 and Figure 5.3 present *precision at 10* and *average precision*, respectively, on preprocessed data. Figure 5.4 shows the *precision at R* for each of the algorithms, where R could vary between 1 and 10.

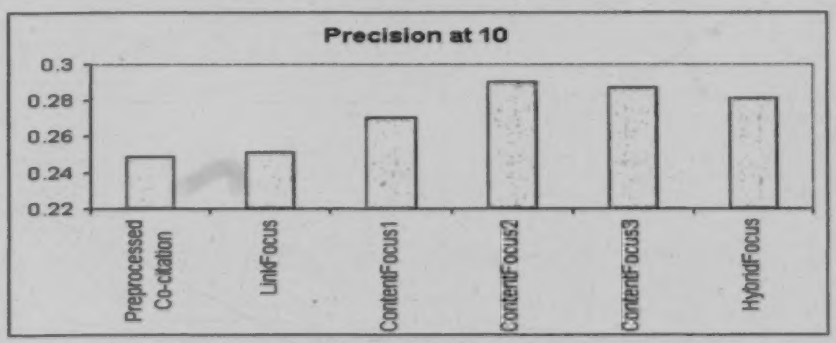


Figure 5.2: Preprocessed data: Precision at 10

We can recognize that the content-based methods consistently outperform Preprocessed Co-citation.

Precision at 10 expresses how many relevant results an algorithm has retrieved, in average, over all the experiments. *Average precision* takes into account also the ranks of the relevant results. LinkFocus returns in average

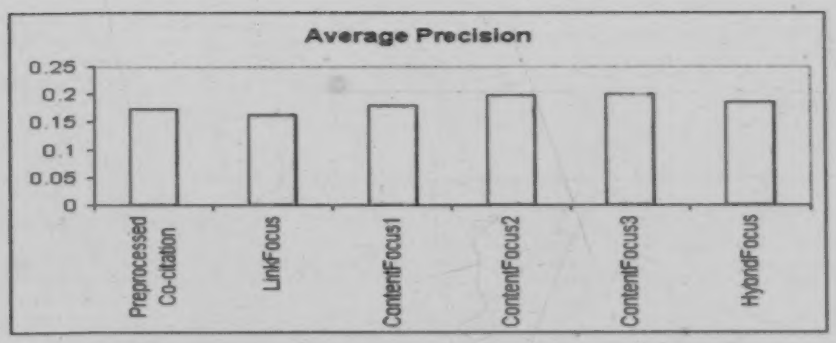


Figure 5.3: Preprocessed data: Average Precision

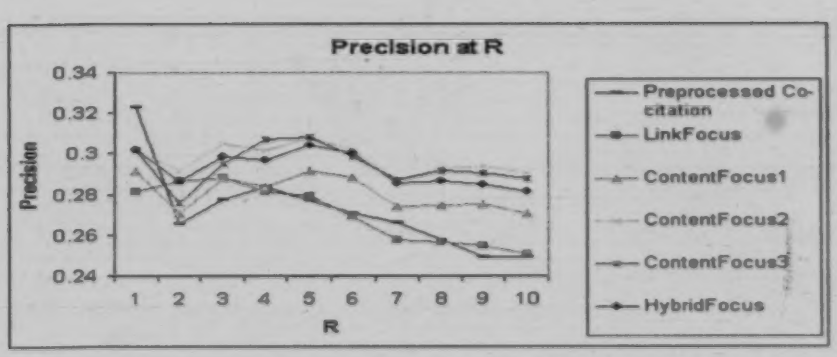


Figure 5.4: Preprocessed data: Precision at R

more relevant results than Preprocessed Co-citation; however, LinkFocus ranks the relevant results lower than Preprocessed Co-citation.

With respect to content, our set of experiments on preprocessed data show that ContentFocus2 and ContentFocus3 perform better than ContentFocus1. ContentFocus2 produces more relevant results than ContentFocus3, but ranks them lower than ContentFocus3. Taking more content into account translates into more keywords extracted for the query URL, as well as into a higher chance of finding these keywords within the text collected for the siblings. This fact does not necessarily imply an increase in the quality of the results. If the number of *informative* keywords increases, then the precision of the results is likely to become higher; on the other hand, if the number of *bogus* keywords increases, the results might be affected. We notice that, on preprocessed data, adding meta description text to title and anchor text is the most effective strategy for content-based focus.

For a given experiment, if the number of keywords extracted for the query URL is at least 5 ($T = 5$ is the empirical threshold from section 4.4), than Hybrid Focus is equivalent with ContentFocus1. For the rest of the cases, the available textual information may not be enough for ContentFocus1 to perform well, so that, overall, HybridFocus is slightly better than ContentFocus1.

We compute the same metrics on the un-preprocessed data.

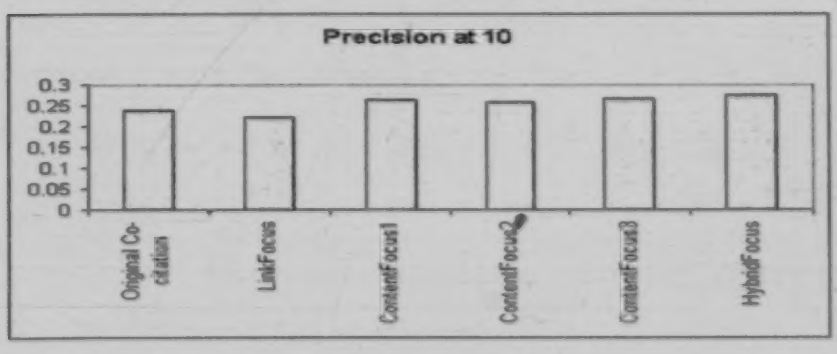


Figure 5.5: Un-Preprocessed data: Precision at 10

We notice that our content-based methods also outperform Original Co-citation, and that HybridFocus performs slightly better than ContentFocus1.

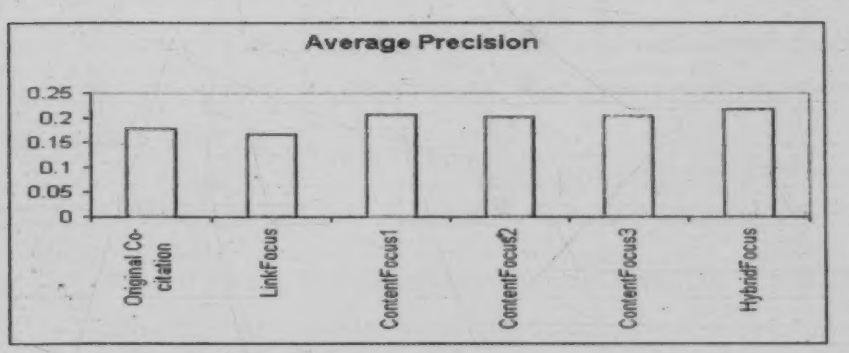


Figure 5.6: Un-Preprocessed data: Average Precision

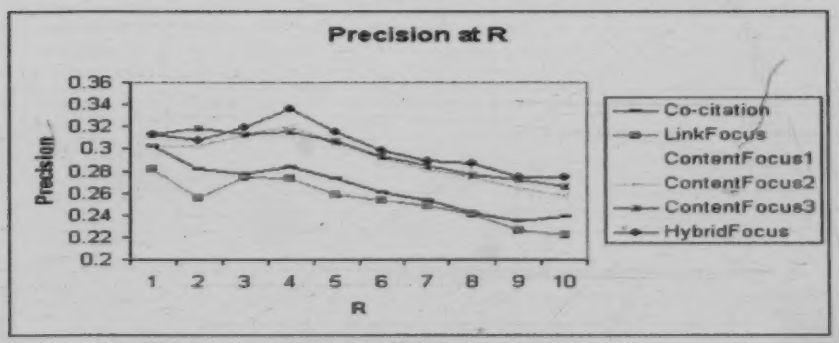


Figure 5.7: Un-Preprocessed data: Precision at R

However, on un-preprocessed data, taking the title and anchor text is the best strategy for content-based focus. The performance of LinkFocus is weaker than the one of Original Co-citation.

We also want to evaluate whether the results returned by our algorithms are generally the same or whether the results are largely disjoint sets of URLs. Table 5.2 illustrates the amount of overlap in the results returned by each pair of algorithms on preprocessed data. Table 5.3 shows the same information, but on un-preprocessed data. The numbers are percentages that indicate the overlap divided by the total number of results returned by the algorithm in that row.

Table 5.2: Preprocessed data: Overlap between methods

	Preprocessed Co-citation	Link Focus	Content Focus1	Content Focus2	Content Focus3	Hybrid Focus
Preprocessed Co-citation	100	68.7	60.2	64.6	65.5	63.2
Link Focus	68.7	100	56.6	61.5	61.7	62.8
Content Focus1	60.2	56.6	100	82.9	81.0	92.4
Content Focus2	64.6	61.5	82.9	100	93.5	83.2
Content Focus3	65.5	61.7	81.0	93.5	100	81.2
Hybrid Focus	63.2	62.8	92.4	83.2	81.2	100

Table 5.3: Un-preprocessed data: Overlap between methods

	Original Co-citation	Link Focus	Content Focus1	Content Focus2	Content Focus3	Hybrid Focus
Original Co-citation	100	82.0	57.8	60.6	61.2	62.0
Link Focus	82.0	100	58.2	60.8	61.5	64.1
Content Focus1	57.8	58.2	100	79.7	77.4	93.0
Content Focus2	60.6	60.8	79.7	100	90.3	78.7
Content Focus3	61.2	61.5	77.4	90.3	100	76.6
Hybrid Focus	62.0	64.1	93.0	78.7	76.6	100

The results in Table 5.2 and 5.3 are similar. HybridFocus and ContentFocus1 have the highest overlap, due to the fact that for most of the experiments, they are equivalent. Content-based methods have a large overlap amongst themselves, indicating that taking increasingly more content into account did not change the results drastically. The large overlap between Preprocessed Co-citation and LinkFocus is not surprising due to the fact that both algorithms exploit in a relatively similar fashion only the linkage structure around the query URL.

According to Table 5.2, we can infer that for many experiments, the number of results returned by both Preprocessed Co-citation and our focused methods is large. We wanted to check how our focused methods perform when we "differ enough" from Co-citation, i.e., when we take into consideration only those experiments for which the intersection of results contains at most 5 elements (we set the threshold 5, meaning that if half of the results are different, than we "differ enough" from Preprocessed Co-citation).

We had 31 experiments that met the threshold. On this set of experiments, we show *precision at 10* (Figure 5.8) and *average precision* (Figure 5.9) on preprocessed data.

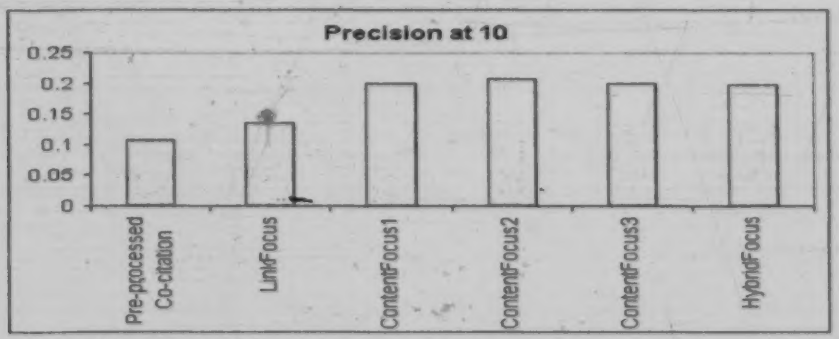


Figure 5.8: Preprocessed Data - Reduced Set: Precision at 10

We can clearly see that our content-based methods outperform Preprocessed Co-citation. LinkFocus obtains more relevant results than Preprocessed Co-citation, but ranks them lower than Preprocessed Co-citation. HybridFocus is similar to ContentFocus1.

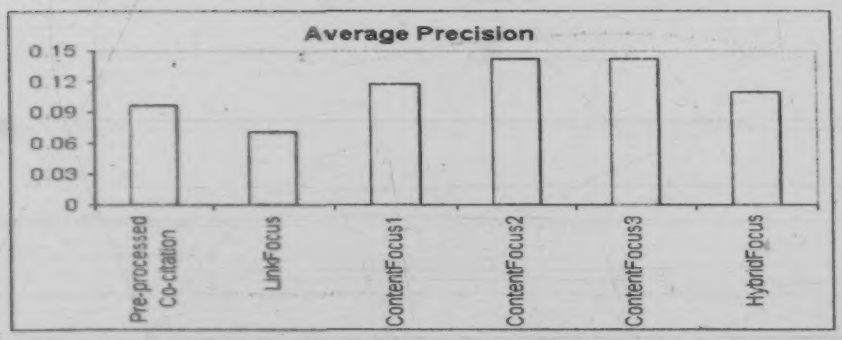


Figure 5.9: Preprocessed Data - Reduced Set: Average Precision

On un-preprocessed data, we had 34 experiments that met the threshold. The results on the reduced set are presented in Figure 5.10 and Figure 5.11.

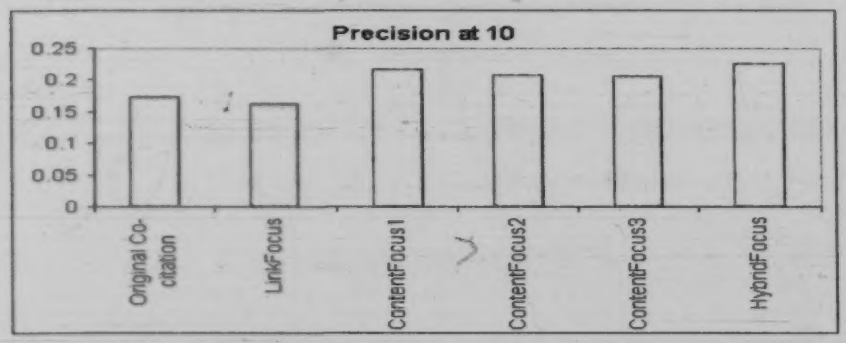


Figure 5.10: Un-Preprocessed Data - Reduced Set: Precision at 10

The same observations as for preprocessed data are valid for un-preprocessed data too.

5.4 Statistical Significance

We also want to evaluate the statistical significance of our results. We have computed the sign test and the Wilcoxon sum of ranks test for pairs of algorithms on both preprocessed and un-preprocessed data.

We are interested to see whether the difference between our focused methods and Co-citation is statistically significant. Table 5.4 shows the results of these statistical tests on preprocessed data.

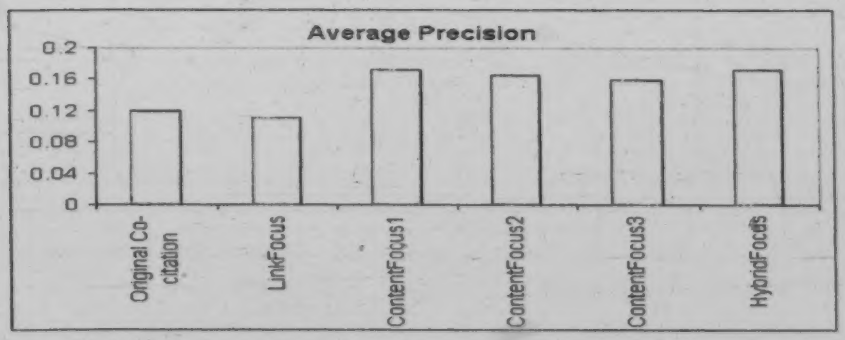


Figure 5.11: Un-Preprocessed Data - Reduced Set: Average Precision

Table 5.4: Preprocessed Data: Sign test and Wilcoxon sum of ranks test

Algorithms	Reduced		All	
	Sign	Rank Sum	Sign	Rank Sum
LinkFocus better than Preprocessed Co-citation	0.14	0.2793	0.4701	0.4685
ContentFocus1 better than Preprocessed Co-citation	0.0006	0.0286	0.0862	0.2033
ContentFocus2 better than Preprocessed Co-citation	0.0002	0.0203	0.0024	0.057
ContentFocus3 better than Preprocessed Co-citation	0.0005	0.0286	0.0043	0.0719
HybridFocus better than Preprocessed Co-citation	0.0009	0.0338	0.0179	0.1103

At the level of significance $\alpha = 0.05$, on the reduced data set, the difference between any of ContentFocus1, ContentFocus2, ContentFocus3, and HybridFocus, respectively, and Preprocessed Co-citation is statistically significant. With respect to the set of all experiments, at the level of significance $\alpha = 0.05$, only the Sign Test indicates a statistically significant difference between any of ContentFocus2, ContentFocus3, and HybridFocus, respectively, and Preprocessed Co-citation. This can be explained by the fact that the reduced set represents one third of the set of all experiments, where the remaining two thirds are those experiments for which our methods and Preprocessed Co-citation have a large overlap, i.e., at least 6 results are shared.

At the level of significance $\alpha = 0.01$, the Sign test indicates a statistically significant difference between any of ContentFocus1, ContentFocus2, ContentFocus3, and HybridFocus, respectively, and Preprocessed Co-citation on the reduced data set, and between any of ContentFocus2, and ContentFocus3, respectively, and Preprocessed Co-citation on the set of all experiments. At this level of significance, Wilcoxon sum of ranks test does not indicate any statistically significant difference.

There is no statistically significant difference between LinkFocus and Preprocessed Co-citation.

We summarize these comments in the Table 5.5 and Table 5.6.

Table 5.5: Preprocessed Data: $\alpha = 0.05$

Algorithms	Reduced		All	
	Sign	Rank Sum	Sign	Rank Sum
ContentFocus1 better than Preprocessed Co-citation	YES	YES		
ContentFocus2 better than Preprocessed Co-citation	YES	YES	YES	
ContentFocus3 better than Preprocessed Co-citation	YES	YES	YES	
HybridFocus better than Preprocessed Co-citation	YES	YES	YES	
LinkFocus better than Preprocessed Co-citation				

We have performed the same computation on un-preprocessed data. Table

Table 5.6: Preprocessed Data: $\alpha = 0.01$

Algorithms	Reduced		All	
	Sign	Rank Sum	Sign	Rank Sum
ContentFocus1 better than Preprocessed Co-citation	YES			
ContentFocus2 better than Preprocessed Co-citation	YES		YES	
ContentFocus3 better than Preprocessed Co-citation	YES		YES	
HybridFocus better than Preprocessed Co-citation	YES			
LinkFocus better than Preprocessed Co-citation				

5.7 shows the results of the sign test and Wilcoxon sum of ranks test. Table 5.6 and Table 5.7 show what methods differ statistically significant at the level of significance $\alpha = 0.05$ and $\alpha = 0.01$, respectively.

Table 5.7: Un-Preprocessed Data: Sign test and Wilcoxon sum of ranks test

Algorithms	Reduced		All	
	Sign	Rank Sum	Sign	Rank Sum
ContentFocus1 better than Original Co-citation	0.0136	0.1003	0.0387	0.1616
ContentFocus2 better than Original Co-citation	0.0503	0.1562	0.1025	0.2385
ContentFocus3 better than Original Co-citation	0.0625	0.1729	0.0311	0.1521
HybridFocus better than Original Co-citation	0.0058	0.0692	0.0071	0.0895
LinkFocus better than Original Co-citation	0.7050	n.a.	0.9247	n.a

We conclude that Focused Co-citation consistently outperforms Co-citation. We have also noticed that Co-citation and our focused methods are slightly better on preprocessed data than on un-preprocessed data. Finally, for the cases when we differ in at least 5 results from Co-citation, we are better than Co-citation. In all other cases, we are at least as good as Co-citation.

Table 5.8: Un-Preprocessed Data: $\alpha = 0.05$

Algorithms	Reduced		All	
	Sign	Rank Sum	Sign	Rank Sum
ContentFocus1 better than Original Co-citation	YES		YES	
ContentFocus2 better than Original Co-citation				
ContentFocus3 better than Original Co-citation			YES	
HybridFocus better than Original Co-citation	YES		YES	
LinkFocus better than Original Co-citation				

Table 5.9: Un-Preprocessed Data: $\alpha = 0.01$

Algorithms	Reduced		All	
	Sign	Rank Sum	Sign	Rank Sum
ContentFocus1 better than Original Co-citation				
ContentFocus2 better than Original Co-citation				
ContentFocus3 better than Original Co-citation				
HybridFocus better than Original Co-citation	YES		YES	
LinkFocus better than Original Co-citation				

5.5 Discussion

The precision values reported above are quite low for all the algorithms. This is, however, not really indicating a weakness of the methods, but is due to the shortcomings of using the Open Directory as "ground truth". Collapsing the tree at level 3 alleviates to some extend these deficiencies. However, many relevant pages found by our algorithms will get a score of 0 simply because they do not appear in the Open Directory at all.

In order to increase the precision of the results, more information about the personal interests of the user has to be taken into account. The precision of our content-based methods depends on the selection of keywords, as we have explained above. For the query URL *www.cs.ubc.ca*, we have assumed all the time that the user is interested in locating other computer science departments of universities from Canada. However, maybe the user is interested in locating information about other departments from the University of British Columbia. The solution is to present the user a list with the keywords that the algorithm has extracted and let him choose the keywords that best fit his interests. The list of keywords extracted by ContentFocus1 for *www.cs.ubc.ca* is presented in Figure 5.12.

additional, british, britishcolumbia, canada, columbia, comp, computa, computer, computi, departamento, department, dept, faculty, information, ncia, ofbritish, sci, science, student, talk, ubc, univ, university

Figure 5.12: Selecting keywords: initial set

Table 5.10 illustrates the effect on ContentFocus1 when the user selects as keywords "computer" and "science", while Table 5.11 is obtained for the keywords "ubc" and "department".

Letting the user to select the keywords of interest has the potential to significantly improve our results, as proved by the above example. We believe that allowing the user to be an active part of the search process could be used for the benefit of the search for information on the Web.

Table 5.10: Selecting keywords: "computer" and "science"

URL	DESCRIPTION
www.ubc.ca	UBC homepage
www.csc.uvic.ca	CS Dept., Univ. of Victoria
www.cs.toronto.edu	CS Dept., Univ. of Toronto
www.csd.uwo.ca	CS Dept., Univ. of Western Ontario
www.cs.umanitoba.ca	CS Dept., Univ. of Manitoba
www.cs.uregina.ca	CS Dept., Univ. of Regina
www.cpsc.ucalgary.ca	CS Dept., Univ. of Calgary
www.cs.usask.ca	CS Dept., Univ. of Saskatchewan
www.scs.carleton.ca	CS Dept., Univ. of Carleton
www.cs.utoronto.ca	CS Dept., Univ. of Toronto

Table 5.11: Selecting keywords: "ubc" and "department"

URL	DESCRIPTION
www.psych.ubc.ca	UBC Psychology Dept.
www.philosophy.ubc.ca	UBC Philosophy Dept.
www.ubc.ca	UBC homepage
www.math.ubc.ca	UBC Mathematics Dept.
www.chem.ubc.ca	UBC Chemistry Dept.
www.zoology.ubc.ca	UBC Zoology Dept.
www.eos.ubc.ca	UBC Earth and Ocean Sciences Dept.
www.stat.ubc.ca	UBC Statistics Dept.
www.microbiology.ubc.ca	UBC Microbiology Dept.
www.botany.ubc.ca	UBC Botany Dept.

5.6 Manual Evaluation

As we discussed before, we need a more reliable strategy for evaluating our results. One alternative would be to perform a systematic user study. We have done a small experiment in which one user (myself) evaluates the results of Preprocessed Co-citation and ContentFocus2 methods (we have picked ContentFocus2 because it is the method with the best performance on preprocessed data). Ten(10) query URLs were chosen at random. With respect to the selection of keywords for ContentFocus2, we have evaluated two alternatives:

- Automatic selection of keywords, as explained in section 4.3
- Manual selection of keywords: the user selects the keywords of interest from the set of automatically generated keywords.

We have evaluated the two algorithms with respect to the topic given by the selected keywords. Figure 5.13 and Figure 5.14 present *precision at 10* and *average precision* for Preprocessed Co-citation and ContentFocus2 when the selection of keywords is done automatically. Figure 5.15 and Figure 5.16 present the same information for the case when the selection is done by the user.

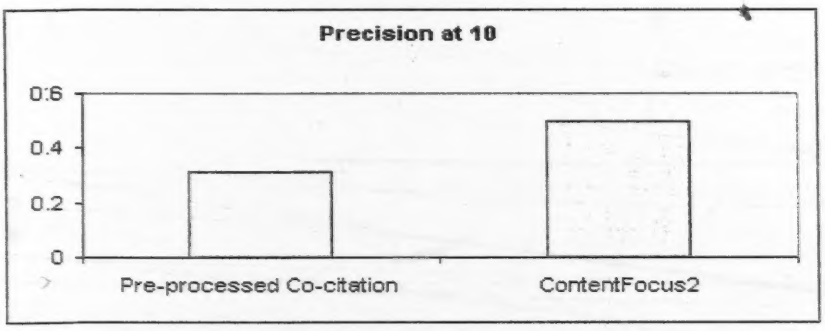


Figure 5.13: Automatic selection of keywords: Precision at 10

These results make us confident that a systematic user study, either with automatic or manual selection of keywords, may be successful in proving the effectiveness of our focused methods.

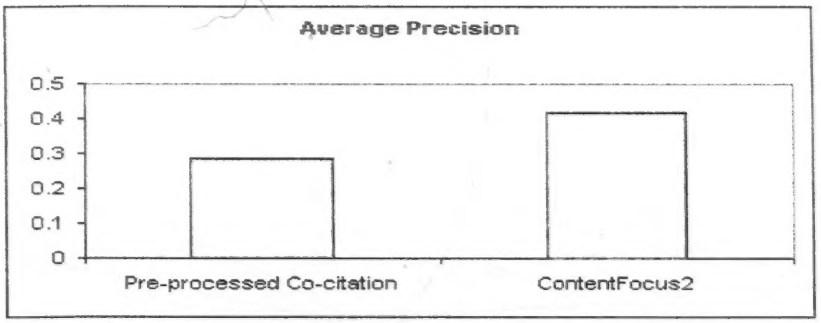


Figure 5.14: Automatic selection of keywords: Average Precision

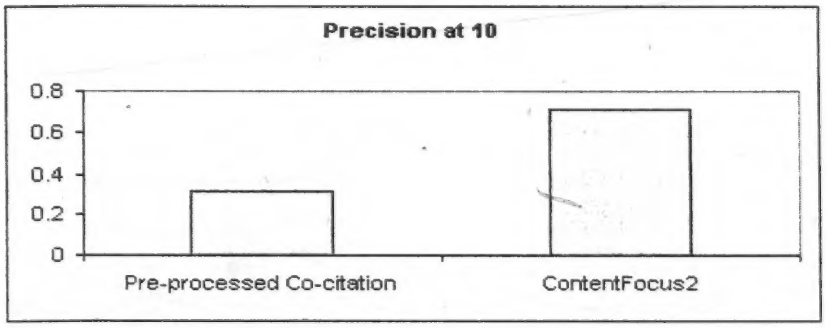


Figure 5.15: Manual selection of keywords: Precision at 10

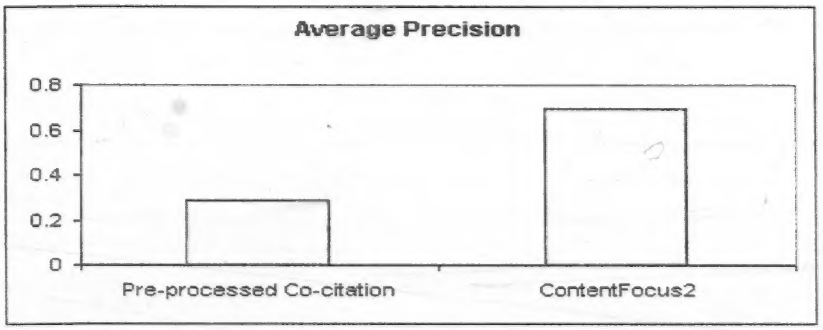


Figure 5.16: Manual selection of keywords: Average Precision

Chapter 6

Conclusions and Future Work

In this thesis we have addressed the problem of finding related pages on the Web. The few papers in the literature that have addressed this problem exploit the linkage structure of the Web and the order of links within Web pages.

We have implemented and analyzed the original Co-citation algorithm [15] and we have identified a number of problems that may affect its results. We have discussed and applied a number of heuristics (navigational links elimination, near-duplicate pages contraction, and pagelet extraction) in order to improve the results of the original Co-citation. We call the succession of these techniques "preprocessing". We have noticed that the effect of preprocessing can not be predicted accurately in all cases, i.e., there are cases when preprocessing helps improving the results of original Co-citation, but there are also cases when preprocessing adversely affects the quality of the results.

We have argued that the problems with Co-citation are mainly due to the existence of "unfocused" pages on the Web. We have formalized a notion of "focus" of a collection of links in several ways, based on content and linkage information. We have embedded our notions of focus within Co-citation and we have shown in the experimental evaluation that our focused versions of Co-citation outperform the unfocused version, both on preprocessed and on un-preprocessed data.

In addition, we propose an interactive search strategy, tailored to the user's personal interests.

6.1 Lessons Learned

Based on the experience developed while working on the problem of finding related pages on the Web, we summarize the following points as "lessons learned":

- Content-based focus is our best performing method. Its precision can be further increased by allowing the user to select the keywords of interest.
- Co-citation and our focused methods are slightly better on preprocessed data than on un-preprocessed data. Shortly, preprocesing does help.
- We need more reliable evaluation strategies than the Open Directory.
- For the cases when Co-citation already works well, the improvement over Co-citation obtained by our methods is not significantly larger.

6.2 Directions for Future Work

There are several possibilities for future research. The evaluation of our algorithms in terms of precision is, as already mentioned above an important issue. To be able to give a more reliable evaluation, we have to look into alternative ways to evaluate such algorithms. An idea would be to compute "coarse", domain-specific, similarity measure that, although far from being definitive or exhaustive, does serve to illustrate important aspects of the proposed algorithms. The precision of our algorithms is low due mainly to the fact that many relevant pages are not classified in the Open Directory at all. We may be able to use an algorithm that, given a Web page, is able to predict with high accuracy the category of the Open Directory where the page should belong to (this project is currently developed at our university). Finally, we do not exclude the possibility of organizing an extensive user study.

Bibliography

[1] Altavista. http://www.altavista.com.

[2] Z. Bar-Yossef and S. Rajagopalan. Template detection via data mining and its applications. In *Proceedings of WWW*, 2002.

[3] K. Bharat and A. Broder. Mirror, mirror on the web: a study of host pairs with replicated content. In *Proceedings of the 8th International WWW Conference*, 1999.

[4] K. Bharat, A. Broder, J. Dean, and M. Henzinger. A comparison of techniques to find mirrored hosts on the www. In *Proceedings of the ACM Digital Library Workshop on Organizing Web Space (WOWS)*, 1999.

[5] K. Bharat and M. Henzinger. Improved algorithms for topic distillation in hyperlinked environments. In *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 1998.

[6] S. Brin and L. Page. The anatomy of a large-scale hypertextual web search engine. In *Proceedings of the 7th International WWW Conference*, 1998.

[7] A. Broder, S. Glassman, M. Manasse, and G. Zweig. Syntactic clustering of the web. In *Proceedings of the 6th International WWW Conference*, 1997.

[8] A. Broder, R. Kumar, F. Maghoul, P. Raghavan, S. Rajagopalan, R. Stata, A. Tomkins, and J. Wiener. Graph structure in the web. In *Proceedings of the 9th International WWW Conference*, 2000.

[9] S. Chakrabarti. Integrating the document object model with hyperlinks for enhanced topic distillation and information extraction. In *Proceedings of WWW*, 2001.

[10] S. Chakrabarti, B. Dom, and P. Indyk. Enhanced hypertext categorization using hyperlinks. In *In Proceedings of the ACM SIGMOD International Conference on Management of Data*, 1998.

[11] S. Chakrabarti, B. Dom, R. Kumar, P. Raghavan, S. Rajagopalan, A. Tomkins, D. Gibson, and J.M. Kleinberg. Mining the link structure of the world wide web. *IEEE Computer*, 32(8), August 1999.

[12] S. Chakrabarti, B. Dom, P. Raghavan, S. Rajagopalan, D. Gibson, and J.M. Kleinberg. Automatic resource compilation by analyzing hyperlink structure and associated text. In *Proceedings of the 7th International WWW Conference*, 1998.

[13] S. Chakrabarti, M. Josji, and V. Tawde. Enhanced topic distillation using text, markup tags, and hyperlinks. In *Proceedings of SIGIR*, 2001.

[14] B. Davison. Recognizing nepotistic links on the web. *Artificial Intelligence for Web Search*, Technical Report WS-00-01:23–28, 2001.

[15] J. Dean and M. Henzinger. Finding related pages in the world wide web. In *Proceedings of the 8th International WWW Conference*, 1999.

[16] Excite. http://www.excite.com.

[17] E. Garfield. Citation indexing. *ISI Press*, 1979.

[18] D. Gibson, J.M. Kleinberg, and P Raghavan. Inferring web communities from link topology. In *Proceedings of the 9th ACM Conference on Hypertext and Hypermedia*, 1998.

[19] G. Golub and C.F. Van Loan. *Matrix Computations*. Johns Hopkins University Press, 1989.

[20] Google. http://www.google.com.

[21] T.H. Haveliwala, A. Gionis, D. Klein, and P. Indyk. Evaluating strategies for similarity search on the web. In *Proceedings of WWW*, 2002.

[22] G. Jeh and J. Widom. Simrank: A measure of structural-context similarity. In *Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2002.

[23] M.M. Kessler. Bibliographic coupling between scientific papers. *American Documentation*, 14:10–25, 1963.

[24] J.M. Kleinberg. Authoritative sources in a hyperlinked environment. *Journal of the ACM*, 46(5):604–632, 1999.

[25] J.M. Kleinberg, R. Kumar, P. Raghavan, S. Rajagopalan, and A. Tomkins. The web as a graph: measurements, models and methods. In *Proceedings of the 5th Annual International Conference on Computing and Combinatorics (COCOON)*, 1999.

[26] R. Kumar, P. Raghavan, S. Rajagopalan, and A. Tomkins. Trawling the web for emerging cyber-communities. In *Proceedings of the 8th International WWW Conference*, 1999.

[27] S. Lawrence. Context in web search. *IEEE Data Engineering Bulletin*, 23(3):25–32, 2000.

[28] A.O. Mendelzon and D. Rafiei. What do the neighbours think? computing web page reputations. *IEEE Data Engineering Bulletin*, 23(3):9–16, 2000.

[29] OCLC Online Computer Library Center Inc., Web Characterization Project. http://wcp.oclc.org.

[30] (ODP) Open Directory Project. http://www.dmoz.org.

[31] L. Page, S. Brin, R. Motwani, and T. Winograd. The pagerank citation ranking: Bringing order to the web. *Stanford Digital Library Technologies Project*, 1998.

[32] P. Pirolli and J. Pitkow. Life, death and lawfulness on the electronic frontier. In *Proceedings of ACM Conference Human Factors in Computing Systems, (CHI)*, 1997.

[33] P. Pirolli, J. Pitkow, and R. Rao. Silk from a sow's ear: Extracting usable structures from the web. In *Proceedings of ACM Conference Human Factors in Computing Systems, (CHI)*, 1996.

[34] D. Rafiei and A.O. Mendelzon. What is the page known for? computing web page reputations. In *Proceedings of the 9th International WWW Conference*, 2000.

[35] U. Shardanand and P. Maes. Social information filtering: algorithms for automating 'word of mouth'. In *Proceedings of the Conference on Human Factors in Computing Systems*, 1995.

[36] Shepard's Citations. http://www.shepards.com.

[37] H. Small. Co-citation in the scientific literature: a new measure of the relationship between scientific documents. *Journal of the American Society for Information Science*, 24:265–269, 1973.

[38] E. Spertus. Mining structural information on the web. In *Proceedings of the 6th International WWW Conference*, 1997.

[39] B. Tadic. Dynamics of directed graphs: the world wide web. *Physica A*, 293(1-2):273–284, 2001.

[40] Teoma. http://www.teoma.com.

[41] B.H. Weinberg. The earliest hebrew citation indexes. *Journal of the American Society for Information Science*, 48(4):318–330, 1997.

[42] Yahoo! http://www.yahoo.com.